INTELLIGENT MATERIAL HANDLING MOBILE ROBOT FOR INDUSTRIAL PURPOSE WITH ACTIVE FORCE CONTROL CAPABILITY

(ROBOT MUDAH GERAK PENGENDALI BAHAN PINTAR UNTUK KEGUNAAN INDUSTRI DENGAN KEUPAYAAN KAWALAN DAYA AKTIF)

MUSA BIN MAILAH
HISHAMUDDIN JAMALUDDIN
ENDRA PITOWARNO
DIDIK SETYO PURNOMO
TANG HOWE HING
MOHAMAD AKMAL BAHRAIN ABDUL RAHIM

RESEARCH VOTE NO:
74016

PUSAT PENGURUSAN PENYELIDIKAN
UNIVERSITI TEKNOLOGI MALAYSIA

2007

UTM/RMC/F/0024 (1998)

UNIVERSITI TEKNOLOGI MALAYSIA

BORANG PENGESAHAN
LAPORAN AKHIR PENYELIDIKAN

TAJUK PROJEK : **INTELLIGENT MATERIAL HANDLING MOBILE ROBOT FOR INDUSTRIAL PURPOSE WITH ACTIVE FORCE CONTROL CAPABILITY**

Saya **MUSA BIN MAILAH**
(HURUF BESAR)

Mengaku membenarkan **Laporan Akhir Penyelidikan** ini disimpan di Perpustakaan Universiti Teknologi Malaysia dengan syarat-syarat kegunaan seperti berikut :

1. Laporan Akhir Penyelidikan ini adalah hakmilik Universiti Teknologi Malaysia.
2. Perpustakaan Universiti Teknologi Malaysia dibenarkan membuat salinan untuk tujuan rujukan sahaja.
3. Perpustakaan dibenarkan membuat penjualan salinan Laporan Akhir Penyelidikan ini bagi kategori TIDAK TERHAD.
4. * Sila tandakan (/)

	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau Kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972).
	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh Organisasi/badan di mana penyelidikan dijalankan).
/	TIDAK TERHAD	

TANDATANGAN KETUA PENYELIDIK

PROFESOR MADYA DR. MUSA MAILAH
Nama & Cop Ketua Penyelidik

Tarikh : **10 APRIL 2007**

CATATAN : ***** *Jika Laporan Akhir Penyelidikan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh laporan ini perlu dikelaskan sebagai SULIT dan TERHAD.*

DEDICATION

To ALL *Intelligent Active Force Control*
(IAFC) group members –
a BIG thank you for your contributions ...

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude and appreciation to the Ministry of Science Technology and Innovation (MOSTI), Malaysia, for providing the project research grant (Project No.: 03-02-06-0038EA067) and the Universiti Teknologi Malaysia (UTM) for their full support in terms of accommodating the research infrastructures and facilities. Special thanks go to Dr. Mohamad Kasim Abdul Jalil from the Department of Design, Abdul Rahim Mohamad, Ahmad Faisal and Ahmad Shahrizam from the Industrial Automation Laboratory for their meaningful contribution in assisting some of the works related to this project. Last but not least to everyone who has in one way or another contributed to the success of this project.

ABSTRACT

This report presents both theoretical and experimental studies of a wheeled mobile system that incorporates a number of intelligent and robust closed-loop control schemes. The system may actually represent an automated material-handling transporter that can be effectively used in a manufacturing or industrial environment. An integrated kinematic and dynamic control with embedded intelligent algorithms were the main approaches employed for the robust motion control of a mobile manipulator (MM) comprising a differentially driven wheeled mobile base platform with a two-link planar arm mounted on top of the platform. The study emphasizes on the integrated kinematic and dynamic control strategy of the schemes in which the former serving as the outer most control loop is used to manipulate the trajectory components while the latter constituting the inner active force control (AFC) loop is implemented to compensate the dynamic effects including the bounded known or unknown disturbances and uncertainties while the system is executing trajectory tracking tasks. The proposed intelligent schemes considered in the study are the fuzzy logic (FL) and knowledge-based system (KBS) strategies that are incorporated into the AFC schemes to automatically estimate the inertia matrix of the system necessary to trigger the disturbance rejection capability. A virtual mobile system was also designed and included in the study to demonstrate the system capability to operate effectively in a virtual computer integrated manufacturing (CIM) environment. The effectiveness and robustness of the proposed schemes were investigated through a rigorous simulation study and later complemented with experimental results obtained through a number of experiments performed on a fully developed working prototype in a laboratory setting. A number of disturbances in the form of vibratory and impact forces are deliberately introduced into the system to evaluate the system performances. The investigation clearly demonstrates the excellent robustness feature of the proposed control scheme compared to other systems considered in the study.

ABSTRAK

Laporan ini mengetengahkan kajian teori dan praktik suatu sistem mudah gerak beroda yang melibatkan beberapa skema pintar dan kawalan gelung tertutup. Sistem ini sebenarnya boleh mewakili suatu pengangkut untuk pemindahan bahan automatik yang boleh digunakan secara berkesan di dalam persekitaran pembuatan atau industri. Kesepaduan kawalan kinematik dan dinamik serta penggunaan algoritma kawalan merupakan pendekatan utama yang digunakan untuk mengawal dengan lasak pergerakan Pengolah mudah gerak (MM) yang terdiri daripada pemacuan differential pelantar mudah gerak beroda dengan lengan dua-penyambung dipasang di atas pelantar tersebut. Kajian memberi penekanan terhadap strategi kesepaduan skema kawalan kinematik dan dinamik di mana bahagian pertama yang wujud di bahagian luar gelung kawalan digunakan untuk mengolah komponen trajektori manakala bahagian kedua melibatkan gelung dalaman kawalan daya aktif (AFC) berrtindak sebagai pemampas kesan dinamik termasuk gangguan terhad diketahui atau tidak dan juga ketidaktentuan semasa sistem beroperasi dalam melaksanakan tugas penjejakan trajektori. Skema pintar yang dicadangkan dalam kajian adalah strategi kawalan logic kabur (FL) dan system berasaskan-pengetahuan (KBS) yang dimuatkan ke dalam AFC untuk menganggarkan matriks inersia sistem secara automatik yang diperlukan untuk menggerakkan keupayaan sistem menghapus gangguan. Suatu sistem mudah gerak maya juga direka bentuk and dimuatkan di dalam kajian untuk mempamerkan kebolehan sistem beroperasi dengan berkesan di dalam persekitaran maya pembuatan komputer bersepadu (CIM). Keberkesanan dan kelasakan skema yang dicadangkan diselidiki menerusi kajian simulasi dan kemudiannya dilengkapi dengan keputusan eksperimen yang diperolehi melalui beberapa eksperimen yang dijalankan terhadap prototaip di dalam persekitaran makmal. Beberapa bentuk gangguan getaran dan hentakan dikenakan pada sistem untuk menilai prestasi system tersebut. Keputusan penyelidikan menunjukkan dengan jelas keupayaan lasak yang baik ditunjukkan oleh skema sistem kawalan cadangan berbanding dengan skema lain yang dipertimbangkan dalam kajian.

.

CONTENTS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

SYMBOL	SUBJECT
$A(q)$	Constraint matrix
$B(q)$	Input transformation matrix
b	Half width of the robot
$C(q,\dot{q})$	*Centripetal* and *Coriolis* matrix
d	the distance of point *G* to *F* of mobile manipulator
$F(q,\dot{q})$	Friction and gravitational vector
g	Acceleration due to gravity (m/s^2)
$G(s)$	A function in *La place* domain representing the feedforward gain in the AFC loop
$G_c(s)$	A function in *La place* domain representing the controller gain
$H(s)$	A function in *Laplace* domain representing the compensated gain in the AFC loop
h	Vector of the *Coriolis* and centrifugal torques
I	Inertia
I'	Estimated inertia
I_m	Motor current
I'_m	Measured motor current
$\boldsymbol{I_c}$	Applied motor current
IN, **IN**	Estimated Inertia matrix
IN_F	Fixed *IN*
IN_I	Integral *IN*
IN_{IF}	Fixed integral *IN*

IN_{init}	Initial *IN*
IN_{IV}	Varied integral *IN*
IN_{KBF}	Knowledge-based fuzzy *IN*
IN_{P}	Proportional *IN*
IN_{PF}	Fixed proportional *IN*
IN_{PV}	Varied proportional *IN*
IN_{RACAFC}	Fixed (crude) *IN* of RACAFC scheme
J	Jacobian
K_p	Proportional constant
K_d	Derivative constant
$K_{pRACAFC}$	Proportional constant for RACAFC scheme
$K_{dRACAFC}$	Derivative constant for RACAFC scheme
K_{tn}	Motor torque constant
$\lambda \in \Re^{r}$	*Lagrange* multiplier
$M(q)$	Symmetric and positive definite inertia matrix
m	Mass
Q	Bounded (known/unknown) disturbance
Q'	Measured disturbance
Q^*	Estimated disturbance
Q_c	Constant torque disturbance
Q_{ca}	Q_c of $[0.2\ 0.2\ 0.5\ -0.5]^T$ Nm
Q_{cb}	Q_c of $[2\ 2\ 5\ -5]^T$ Nm
Q_{cc}	Q_c of $[20\ 20\ 20\ -20]^T$ Nm
Q_{cd}	Q_c of $[30\ 30\ 30\ -30]^T$ Nm
Q_{gain}	Scaling factor for impact and vibration
Q_{imp}	Impact disturbance
Q_{vib}	Vibration disturbance
$q \in \Re^{p}$	p generalized coordinate
r	radius of wheel
$S(q)$	Transformation matrix
θ	Angular position
$\dot{\theta}$	Angular velocity

$\ddot{\theta}$	Angular acceleration
$\ddot{\theta}$	Measured angular acceleration
θ_{ref}	Reference angular position
$\dot{\theta}_{ref}$	Reference angular velocity
$\ddot{\theta}_{ref}$	Reference angular acceleration
θ_{act}	Actual angular position
$\dot{\theta}_{act}$	Actual angular velocity
$\ddot{\theta}_{act}$	Actual angular acceleration
τ	Torque
T_q	Applied torque
φ	Heading angle of platform

LIST OF ABBREVIATIONS

ADC	Analog to Digital Converter
AFC	Active Force Control
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
CIM	Computer Integrated Manufacturing
CPU	Central Processing Unit
D	Derivative
DAC	Digital to Analog Converter
DAS	Data Acquisition System
DC	Direct Current
DDMR	Differentially Driven Mobile Robot
DOF	Degree of Freedom
DOS	Disk Operating System
FL	Fuzzy Logic
FS	Fuzzy System
I	Integral
IL	Iterative Learning
ILC	Iterative Learning Control
KBF	Knowledge Based Fuzzy
KBFAFC	Knowledge Based Fuzzy Active Force Control
KBS	Knowledge Based System
MF	Membership Function
MM	Mobile Manipulator
MMAFCON	Mobile Manipulator Active Force Control Online
NN	Neural Network
P	Proportional

PC	Personal Computer
PD	Proportional plus Derivative
PI	Proportional plus Integral
PIC	Peripheral Interface Controller
PID	Proportional plus Integral plus Derivative
PLC	Programmable Logic Controller
RAC	Resolved Acceleration Control
RACAFC	Resolved Acceleration Control - Active Force Control
RACKBFAFC	Resolved Acceleration Control - Knowledge Based Fuzzy Active Force Control
RMAC	Resolved Motion Acceleration Control
RMRC	Resolved Motion Rate Control
RTM	Real-Time Monitor
RTW	Real-Time Workshop
SMI	Small and Medium-sized Industries
TTE	Trajectory Tracking Error
VE	Virtual Environment
VR	Virtual Reality
WMR	Wheeled Mobile Robot
WTK	WorldToolKit

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 General Introduction

The mobile manipulator is basically a conventional robotic arm mounted on a moving base. It is analogous to a human being considering the body and arm sections. He or she must control these parts integrally when performing a dynamic task to achieve the desired performance. As an example a welder carrying out a welding operation needs to carry out the task by coordinating (control) simultaneously and continuously both the arm and body movements so that a favorable outcome could be obtained from executing the task. It is indeed more challenging when the human operators are replaced by robots or automated machines. It is desirable that the system is equipped with intelligence and robust control capability so that it can perform its tasks satisfactorily and according to specific requirements. Thus, it has to be designed and developed taking into account the aforesaid criteria. Subsequently, comprehensive analytical studies on the kinematics, dynamics and control aspects of the physical system should be carefully carried out in order to come up with an alternative system that produces results comparable to those of human operators. It is useful to note that the mobile transporter described throughout this report shall be synonymously known as wheeled mobile robot (WMR) or mobile manipulator.

1.2 Outline of Research

The use of an autonomous mobile robot as a material handling robot (transporter) is common in the manufacturing and industrial sector including the *Small and Medium-sized Industries* (SMI). This implementation of such a system is a well known fact in view of its ability to replace the operator's (human) inconsistency and shortcoming for repetitive, labour intensive, arduous and continuous tasks. The mobile robot used in this context is usually known as *Automated Guided Vehicle* (AGV). The AGV system has however severe limitations in terms of intelligence (in decision making) while navigating and usually prone to errors whenever it is subject to obstacles or disturbances in its path. In addition to that industrial AGV normally transports passively the work piece or material within the plant layout on the shop floor. This project involves the design and development of a new, innovative, intelligent and robust mobile robot as a transporter with the incorporation of a novel control scheme employing *Active Force Control* (AFC) strategy. A very useful feature of the proposed system is the presence of a dexterous robotic arm that can manipulate positively (via gripper) parts or materials to be transported. It is expected that in the process of manipulating work pieces or parts dynamically by maneuvering the system along a specific guided path, the transporter is subject to a number of so called 'disturbances and uncertainties'. These may be in the form of obstacles in its path, different payloads, complexity of the shape of the parts, compliancy with the environment, vibration of the system, unexpected loading condition, friction (internal or external) and others. The practical AFC scheme has been shown to demonstrate superior robust and effective all round performance in encountering and compensating the disturbances and uncertainties prevalent in dynamical systems. At the same time, intelligent mechanism shall be incorporated into the system to make appropriate decisions when the robot encounters a number of conditions while performing its task. It is the aim of the project to come up with a practical mobile robotic system with real-time active force control scheme which has the quality and capability of handling and transporting the materials robustly, smoothly and efficiently. Sensory devices are appropriately installed at strategic location on the system to provide the essential feedback mechanism. Besides, a vision system is also used to enhance the intelligence of the system. A total mechatronic approach is

applied in the design and development of the system involving the practical integration of mechanical, electrical/electronic and computer control.

1.3 Objective of Research

The specific objective of research is to design and develop fully a working prototype of an intelligent and autonomous material handling mobile robot (WMR or mobile manipulator) for use in an SMI environment.

1.4 Scope of Research

The scope of research shall encompass the followings:

- Literature review on mobile manipulator modelling, motion control and intelligent systems applied to robot control.

- The mobile robot is of a differentially driven wheeled type. Motion and path planning of the mobile robot shall be investigated and applied considering only nonholonomic constraints.

- Theoretical design and simulation of a number of advanced robust motion control schemes based on Active Force Control (AFC) with intelligent elements, namely fuzzy logic (FL) and knowledge-based system (KBS). The application of a virtual reality technique shall also be explored in the research. The main software design tool is MATLAB/Simulink.

- Design and development of an experimental mobile manipulator including mechatronics design of the robot, PC-based interfacing, sensors and actuators interfacing and a robot controller programming using C language. A hardware-

in-the-loop system based on MATLAB/Simulink and Real-Time Workshop (RTW) shall be rigorously experimented. The possibility of using a microcontroller (wireless and embedded system) and digital camera as a vision system shall also be explored. The testing will be done in a laboratory setting that emulates a factory layout.

1.5 Research Methodology

The project will be divided into five main parts. These are modelling and simulation using computer, intelligent controller design, design and development of the prototype, experimentation and performance evaluation and analysis. Initially these areas will be investigated separately by groups of researchers under the supervision of a project leader. Ideas and findings by all the groups will be presented to other group members for transfer of knowledge and experience. The more detailed description of the research methodology is as follows:

i. Computer modelling and simulation

The modelling and simulation phase involves rigorous study of mathematical equations representing the system's kinematics and dynamics. The modelling will take into account realistic and valid assumptions related to the physical systems such as the manipulator and steering or navigation mechanism. Classical control plus intelligent techniques using FL and KBS will be studied independently and later implemented with the AFC strategy to control the system. Their algorithms and techniques will be thoroughly investigated and studied. Comprehensive simulation works will include the evaluation of the system's performance and robustness against various forms of disturbances and other different operating conditions. Changes in the parameters such as those related to the physical arm, simulation and learning algorithms will also be also taken into account. Comparative study between the control strategies will also be considered to provide a useful platform in determining the optimum control method. The simulation works serve as a basis for designing and developing the prototype in later stages.

ii. Intelligent controller design

A suitable intelligent controller will be identified based on the outcome of the simulation study. The controller is chosen such that its practical implementation in real time is feasible and readily applied. Relevant parameters of the controller will be refined and related computer program will be written for this purpose. Development of a possible hardware for the controller will be investigated and looked into. This will largely involve aspects of computer programming, interfacing, electrical/electronics and embedded system technology.

iii. Design and development of the prototype

The design and development of the proposed mobile robotic system is based on total *Mechatronic* approach. This involves the integration of a number of classical engineering disciplines namely, the mechanical, electrical/electronics and computer-based control.

Mechanical: Critical design process of the mechanical aspect should be initiated to come up with the best ever possible product design of the robot. The mechanical system should comprise the base structure, wheel mechanisms, transmission drive system, main body, chassis, mounting, work table and a manipulator with special gripper attachment at the end of the arm. The design process will include conceptual and final design of the robot involving the development suitable mechanisms, static and dynamic analysis of the structure of the system, selection of materials and others which are all referred to and should comply with the pre-determined design criterion. The design of the wheel mechanisms are of particular importance in view of the mobility and maneuverability aspects. The design should also take into account the ease of the fabrication of the parts to be processed.

Electrical/electronics: The selection of the drive and actuator system should be based on the force analysis of the system to ensure proper actuation of the system is achieved. DC servo motors with suitable transmission drives are expected to be used in the system. The actuators should be driven by a set of suitable power

packed portable batteries. The essential electronics involve the incorporation of the motor drivers, sensory and signal conditioning devices. A number of sensors are used and placed strategically in the system. Inductive sensor is installed to provide sensing signal related to the guidance system. Another additional feature to be incorporated in the system is the vision system involving the use of CCD camera and image processing software The AFC scheme specifically uses a current sensor and accelerometer to obtain the measured values of the parameter necessary for the control action. The signal conditioning aspect utilizes a number of circuits such as those related to amplification/attenuation, buffer, active filters, and analog-to-digital (A/D) and digital-to-analog (D/A) conversions.

Computer-based control: The microprocessor-based control will ultimately implement the embedded system technology. Initial work will involve the use of PC-based devices and software to control the system. At a later stage, this will be transformed to wireless and fully autonomous system with suitable embedded components.

iv. Experimentation

Upon complete integration of the system components and hence the working prototype of the WMR, experimentation will be rigorously carried out to test the effectiveness of the proposed system. The tests will take into account various operating and loading conditions.

v. Performance evaluation and analysis

Finally the system's performance will be critically evaluated and analysed. This includes the results obtained from both the simulated and prototype developmental activities. The research outcomes should provide insights and information which would be useful for future development, improvement and expansion of the system. In addition to that, suggestion for further research works will also be outlined.

1.6 Organization of Report

The report is organized into eight chapters. In Chapter 2, the fundamental concepts, underlying theories and reviews of the main topics of research pertaining to modelling of the kinematics and dynamics of the mobile manipulator and their control, resolved-motion-rate-control (RMRC) and resolved acceleration control (RAC), active force control (AFC), fuzzy logic (FL) and knowledge-based system (KBS), motion planning and virtual reality (VR) are described. The modelling of the mobile robot systems comprising the platform and arm is first described as this provides the basis for the simulation of the control system. Next, the principles of the well known RAC and the pure AFC method is discussed with special attention focused on the method to enhance the strategy using intelligent means, namely FL and KBS methods. The KBS inference mechanism, i.e. knowledge investigation and validation, knowledge representation, knowledge acquisition and knowledge processing are discussed as well as the KBS procedures. A preliminary discussion on the use of knowledge-based method to a fuzzy system is also addressed. A VR technique is highlighted and introduced in this section.

Chapter 3 is on $v\omega$-AFC with FL method. In this chapter, an inertia matrix estimation method using fuzzy logic for an AFC scheme is presented. The fuzzy logic mechanism computes the estimated inertia matrix (**IN**) automatically and continuously, while the mobile robot is tracking the reference trajectory. The performance of the proposed $v\omega$-AFC-fuzzy logic control method is compared to the $v\omega$-AFC control scheme which has adopted the crude approximation technique in the estimation of **IN**.

Chapter 4 elaborates the application of a VR technique applied to an AFC method for the motion planning and control of a WMR in computer integrated manufacturing (CIM) setting**.** The first section of the chapter touches on a description of the robot simulator and highlights the overall architecture of the proposed virtual WMR simulator. This is followed by the construction of the scene graph for the virtual environment (VE) of the mobile robot's workspace (layout). In addition, the elements within the scene graph will be also briefly discussed. The simulation study

based on some of the theoretical concepts and principles derived from previous chapters are fully described in this chapter. On top of that, a collision avoidance algorithm is included in the chapter for the intelligent motion path planning of the system.

Chapter 5 describes a simulation study of the new proposed scheme known as *Resolved Acceleration Control and Active Force Control* (RACAFC) which employs a crude approximation on the inertia matrix estimation. This proposed scheme is considered as the basic robust motion control using AFC applied to the mobile manipulator that deals with the integration of the robot's kinematics and dynamics. The tuning procedure of the inertia matrix estimator for both the mobile platform as well as for the manipulator is rigorously discussed in this chapter. Some disturbances were introduced to test the robustness of the proposed scheme.

Chapter 6 presents a simulation study of the main proposed scheme, i.e., *Resolved Acceleration Control and Knowledge-Based Fuzzy Active Force Control* (RAC-KBFAFC). The complete procedure to realize the knowledge-based fuzzy including the procedures of knowledge investigation, validation, representation, acquisition and processing is sufficiently discussed. The procedure to investigate the knowledge through appropriate acquisition process is highlighted. The knowledge shall be used as the reasoning to design the proper fuzzy output function in the development of the fuzzy rules inference mechanism applied to estimate the **IN** of the system. This chapter also presents the complete results of the simulation study of RAC-KBFAFC scheme that is subjected to a number of conditions.

Chapter 7 describes the mechatronic design and development of the experimental mobile manipulator in the form of a working prototype comprising a differentially driven mobile robot/platform with a two-link planar robot arm mounted on the top of the platform. The system is equipped with graphical and real-time monitor (RTM) control programming feature. This chapter also elaborates the programming and experimental procedure based on the RAC, RACAFC and RACKBFAFC schemes. The experimentation shall provide a basis for validating the theoretical (simulation) counterpart.

Finally, Chapter 8 concludes the research project. The directions and recommendations for future research works are also outlined in this chapter. Some of the additional materials pertaining to relevant algorithms, computer programs or modules, publications and achievements related to the research are enclosed in the appendices.

CHAPTER 2

THEORETICAL FRAMEWORKS AND REVIEW

2.1 Introduction

This chapter is concerned with the description of various theoretical concepts and literature that are directly related to the undertaken research. These are specifically stated as the kinematic and dynamic and modelling of the mobile manipulator and their control, resolved-motion-rate control (RMRC), resolved acceleration control (RAC), active force control (AFC), fuzzy logic (FL), knowledge-based system (KBS), motion path planning and virtual reality (VR) technique. The proposed AFC-based schemes that are described in succeeding chapters both for the simulation and experimental works shall utilize in parts or full, the various concepts outlined in this chapter through suitably designed procedure and implementation.

2.2 Mathematical Modelling of Mobile Manipulator

The modelling of the mobile manipulator is important prior to any simulation work that may include the control element. This shall be duly described in the subsequent sections in terms of the dynamic and kinematic aspects.

2.2.1 Dynamics of Mobile Manipulator

Recall the dynamics of a mobile manipulator obtained using the *Lagrangian* approach that is subjected to kinematic constraints in the form (Lin, 2001):

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(q,\dot{q}) + A^T(q)\lambda + \tau_d = B(q)\tau \qquad (2.1)$$

where the kinematic constraints are subjected to:

$$A(q)\dot{q} = 0 \qquad (2.2)$$

$q \in \Re^p$ represents the *p* generalized coordinate, $M(q) \in \Re^{p\times p}$ is a symmetric and positive definite inertia matrix, $C(q,\dot{q}) \in \Re^{p\times p}$ denotes centripetal and *Coriolis* matrix, $F(q,\dot{q}) \in \Re^p$ is friction and gravitational vector, $A(q) \in \Re^{r\times p}$ is a constraint matrix, $\lambda \in \Re^r$ is the Lagrange multiplier which represents the vector of constraint forces, $\tau_d \in \Re^p$ is a bounded unknown disturbances including unstructured dynamics, $B(q) \in \Re^{p\times(p-r)}$ is the input transformation matrix, and $\tau \in \Re^{p-r}$ represents torques input vector. If generalized coordinates *q* are described into two sets, i.e. q_v of the mobile platform and q_r of the manipulator, $q = \left(q_v^T, q_r^T\right)^T$ where $q_v \in \Re^m$ are the generalized coordinates in the kinematic constraints of Equation (2.2), and $q_r \in \Re^n$ are the generalized coordinates of the local manipulator system without kinematic constraints, then Equation (2.2) can be simplified into:

$$A_v(q_v)\dot{q}_v = 0 \text{ with } A_v(q_v) \in \Re^{r\times m}, \qquad (2.3)$$

with *r* is the kinematic constraints.

The following properties hold in Equation (2.1) (Lewis *et al*., 1999):

Property 2.1 (*Parameter Boundedness*):

$$M_{\min} I_p \le M(q) \le M_{\max} I_p$$
$$C(q,\dot{q}) \le C_b(q)\|\dot{q}\| \tag{2.4}$$

where $M_{\min}$ and $M_{\max}$ are the positive scalar constants that are dependent to the robot's mass properties and constraint matrix, I_p is a $p \times p$ identity matrix, and $C_b(q)$ is a positive function of q. In this thesis all of the joints are assumed to be revolute so that $C_b(q)$ is a finite positive constant (Lewis *et al*., 1993).

Property 2.2 (*Skew Symmetric*):

$$\dot{M} - 2C = -(\dot{M} - 2C)^T$$
$$\dot{M} = C + C^T \tag{2.5}$$

By implementing Equation (2.3) and separating the generalized coordinates into two sets, $q = \left(q_v^T, q_r^T\right)^T$ Equation (2.1) can be rewritten as

$$\begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}\begin{pmatrix} \ddot{q}_v \\ \ddot{q}_r \end{pmatrix} + \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}\begin{pmatrix} \dot{q}_v \\ \dot{q}_r \end{pmatrix} + \begin{pmatrix} F_v \\ F_r \end{pmatrix} + \begin{pmatrix} A_v^T(q_v)\lambda \\ 0 \end{pmatrix} + \begin{pmatrix} \tau_{dv} \\ \tau_{dr} \end{pmatrix} = \begin{pmatrix} B_v \tau_v \\ \tau_r \end{pmatrix} \tag{2.6}$$

where $\tau_v \in \Re^{m-r}$ is the actuated torque vector of the constrained coordinates that subject to nonholonomic constraints, $B_v \in \Re^{m\times(m-r)}$ is input transformation matrix of the mobile platform, τ_{dv}, τ_{dr}, F_v, and F_r are torque disturbances and frictions and gravity vector for mobile platform and manipulator respectively. The properties related to Equation (2.6) can then be expressed as:

Property 2.3 (*Skew Symmetric*):

$$\dot{M}_{11} - 2C_{11} = -(\dot{M}_{11} - 2C_{11})^T$$
$$\dot{M}_{22} - 2C_{22} = -(\dot{M}_{22} - 2C_{22})^T \quad (2.7)$$

Property 2.4:

$$\dot{M}_{21} = C_{21} + C_{12}^T$$
$$M_{12} = M_{21}^T \quad (2.8)$$

By considering Equation (2.3) as the equation subject to the standard matrix theory a full rank transformation matrix $S_v(q_v) \in \Re^{m\times(m-r)}$ which spans the null space of $A_v(q_v)$ can be defined as

$$S^T(q_v)A_v^T(q_v) = 0 \quad (2.9)$$

where $S(q_v)$ is bounded at $S(q_v) \le \varsigma$ with ς is positive number. Note that $A_v(q_v)$ is bounded. From Equation (2.9), an auxiliary vector of the platform velocity $v(t) \in \Re^{m-r}$ can be obtained in the form

$$\dot{q}_v = S(q_v)v(t) \quad (2.10)$$

then, in the form of acceleration mode or its derivative is given by:

$$\ddot{q}_v = S(q_v)\dot{v} + \dot{S}(q_v)v \quad (2.11)$$

From Properties 2.1 and 2.2 and by considering Equations (2.1) and (2.11), additional properties can be defined (Lin and Goldenberg, 2002) as follows:

Property 2.5:

$$\begin{aligned} \left\| S^T M_{11} S \right\| &\le \overline{M}_{11b} \left\| M_{12} \right\| \le M_{12b} \\ \left\| M_{21} \right\| &\le M_{21b} \left\| M_{22} \right\| \le M_{22b} \end{aligned} \tag{2.12}$$

where $\overline{M}_{11b}$, M_{12b}, M_{21b}, M_{21b} are positive constants that are assumed to be unknown.

Property 2.6:

$$\begin{aligned} \left\| S^T C_{11} S \right\| &\le \overline{C}_{11b} \left\| \dot{q} \right\| \left\| C_{12} \right\| \le C_{12b} \left\| \dot{q} \right\| \\ \left\| C_{22} \right\| &\le C_{22b} \left\| \dot{q} \right\| \left\| C_{21} \right\| \le C_{21b} \left\| \dot{q} \right\| \end{aligned} \tag{2.13}$$

where $\overline{C}_{11b}$, C_{12b}, C_{21b}, C_{21b} are positive constants.

Generally, Equations (2.6) and Equation (2.10) describe the dynamic equations of the mobile manipulator that subject to kinematic constraints. These are used to design the proposed motion control of the robot system that deals with the kinematics and dynamics.

2.2.2 Kinematics of Mobile Manipulator

Figure 2.1 (a) shows the mobile platform moves by driving the two independent wheels (left and right). In the figure, *XY* is the world coordinate system, *xy* is the robot coordinate system, φ is the robot's heading angle, 2*b* is the width of the robot, *r* is the radius of the wheels, and *d* is the distance of point *G* to *F*. It is assumed that the velocity at which this system moves is relatively slow and therefore the two driven wheels do not slip sideways. The velocity of the platform center of

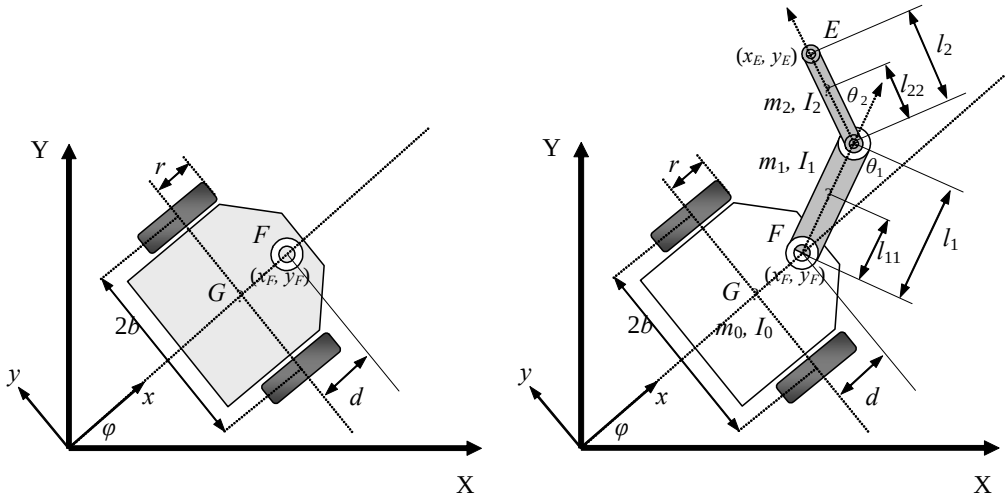
mass, v_G, is then perpendicular to the wheel axis. This expresses x and y components in a nonholonomic equation as follows:

$$\dot{x}_G \sin\varphi - \dot{y}_G \cos\varphi = 0 \tag{2.14}$$

For point F, the constraint can be written as:

$$\dot{x}_F \sin\varphi - \dot{y}_F \cos\varphi + \dot{\varphi}\ d\ = 0 \tag{2.15}$$

where d is the distance between G and F.



(a) (b)

Figure 2.1: (a) A mobile platform and (b) A mobile manipulator

Following Papadopoulos and Poulokakis (2000) and considering Equation (2.15), the general coordinate system, $q = [x_F, y_F, \varphi]^T$ of the robot assuming that its wheels roll and do not slip, the direct kinematic can be expressed as:

$$\begin{pmatrix} \dot{x}_F \\ \dot{y}_F \\ \dot{\varphi}_F \end{pmatrix} = S_v(q_v) \begin{pmatrix} \dot{\theta}_L \\ \dot{\theta}_R \end{pmatrix} \quad \text{or} \quad \dot{q}(t) = S_v(q_v)\dot{\theta}(t) \tag{2.16}$$

and its inverse kinematic is:

$$\dot{\theta}(t) = S_v^{-1}(q)\dot{q}(t) \tag{2.17}$$

where $S_v(q_v)$ is derived from Figure 2.1 (a) as:

$$S_v(q_v) = \begin{pmatrix} \frac{r}{2}\cos\varphi + \frac{d\cdot r}{2b}\sin\varphi & \frac{r}{2}\cos\varphi - \frac{d\cdot r}{2b}\sin\varphi \\ \frac{r}{2}\sin\varphi - \frac{d\cdot r}{2b}\cos\varphi & \frac{r}{2}\sin\varphi + \frac{d\cdot r}{2b}\cos\varphi \\ -\frac{r}{2b} & \frac{r}{2b} \end{pmatrix} \tag{2.17}$$

and $\dot{\theta}_L$, $\dot{\theta}_R$ are the angular velocities of the left and right wheel respectively.

Equation (2.16) can be rewritten as:

$$\begin{pmatrix} \dot{x}_F \\ \dot{y}_F \\ \dot{\varphi}_F \end{pmatrix} = \begin{pmatrix} \frac{r}{2}\cos\varphi + \frac{d\cdot r}{2b}\sin\varphi & \frac{r}{2}\cos\varphi - \frac{d\cdot r}{2b}\sin\varphi \\ \frac{r}{2}\sin\varphi - \frac{d\cdot r}{2b}\cos\varphi & \frac{r}{2}\sin\varphi + \frac{d\cdot r}{2b}\cos\varphi \\ -\frac{r}{2b} & \frac{r}{2b} \end{pmatrix} \begin{pmatrix} \dot{\theta}_L \\ \dot{\theta}_R \end{pmatrix} \tag{2.18}$$

Equation (2.18) can be expressed in the form of rotation matrix as follows:

$$\begin{pmatrix} \dot{x}_F \\ \dot{y}_F \end{pmatrix} = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} \frac{r}{2} & \frac{r}{2} \\ -\frac{d\cdot r}{2b} & \frac{d\cdot r}{2b} \end{pmatrix} \begin{pmatrix} \dot{\theta}_L \\ \dot{\theta}_R \end{pmatrix} \tag{2.19}$$

For the manipulator serially mounted on board the platform at point *F* as in Figure 2.1 (b), its forward kinematic can be described as:

$$\begin{pmatrix} \dot{x}_E \\ \dot{y}_E \end{pmatrix} = \begin{pmatrix} \dot{x}_F \\ \dot{y}_F \end{pmatrix} + \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix} \begin{pmatrix} \dot{\theta}_1 + \dot{\varphi} \\ \dot{\theta}_2 \end{pmatrix} \tag{2.20}$$

where

$$J_{11} = -l_1 \sin\theta_1 - l_2 \sin(\theta_1 + \theta_2) \tag{2.21}$$

$$J_{12} = -l_2 \sin(\theta_1 + \theta_2) \tag{2.22}$$

$$J_{21} = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \tag{2.23}$$

$$J_{22} = l_2 \cos(\theta_1 + \theta_2) \tag{2.24}$$

Equation (2.20) indicates that the kinematic control of the two sub-systems (platform and manipulator) can be partially solved. It is sometimes very useful to analyze the singularity of the system when the robot arm is out of reach beyond its workspace. If (x_F, y_F) is assumed to be in fixed position (platform is not moving and hence $\dot{\varphi}$ = 0), Equation (2.20) can be expressed as:

$$\begin{pmatrix} \dot{x}_T \\ \dot{y}_T \end{pmatrix} = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} \tag{2.25}$$

where (x_T, y_T) is the tip position coordinate relative to the workspace of the manipulator. Then its inverse kinematic is:

$$\begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix}^{-1} \begin{pmatrix} \dot{x}_T \\ \dot{y}_T \end{pmatrix} \tag{2.26}$$

Substituting Equations (2.19), (2.21), (2.22), (2.23) and (2.24) into Equation (2.20) and letting $q = [x_F, y_F, x_E, y_E]^T$ as the input reference coordinate, the complete direct kinematic equation of the mobile manipulator is given by:

$$\begin{bmatrix} \dot{x}_E \\ \dot{y}_E \\ \dot{x}_F \\ \dot{y}_F \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & 0 \\ \sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & \cos\varphi & -\sin\varphi \\ 0 & 0 & \sin\varphi & \cos\varphi \end{bmatrix}$$

$$\begin{bmatrix} \frac{r}{2} - J_{11}\frac{r}{b} & \frac{r}{2} + J_{11}\frac{r}{b} & J_{11} & J_{12} \\ -(d + J_{21})\frac{r}{b} & (d + J_{21})\frac{r}{b} & J_{21} & J_{22} \\ \frac{r}{2} & \frac{r}{2} & 0 & 0 \\ -d\frac{r}{b} & d\frac{r}{b} & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_L \\ \dot{\theta}_R \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \tag{2.27}$$

where $\dot{\theta}_L$ and $\dot{\theta}_R$ are the angular velocities of the left and right wheels respectively, $\dot{\theta}_1$ and $\dot{\theta}_2$ are the angular velocities of the joint angle of link-1 and link-2 respectively.

Letting a full rank transformation matrix $S(q)$ in Equation (2.27) as:

$$S(q) = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & 0 \\ \sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & \cos\varphi & -\sin\varphi \\ 0 & 0 & \sin\varphi & \cos\varphi \end{bmatrix}$$

$$\begin{bmatrix} \frac{r}{2} - J_{11}\frac{r}{b} & \frac{r}{2} + J_{11}\frac{r}{b} & J_{11} & J_{12} \\ -(d + J_{21})\frac{r}{b} & (d + J_{21})\frac{r}{b} & J_{21} & J_{22} \\ \frac{r}{2} & \frac{r}{2} & 0 & 0 \\ -d\frac{r}{b} & d\frac{r}{b} & 0 & 0 \end{bmatrix} \tag{2.28}$$

the direct kinematic equation can be expressed as:

$$\begin{bmatrix} \dot{x}_E \\ \dot{y}_E \\ \dot{x}_F \\ \dot{y}_F \end{bmatrix} = S(q) \begin{bmatrix} \dot{\theta}_L \\ \dot{\theta}_R \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \tag{2.29}$$

and its inverse kinematic is:

$$\begin{bmatrix} \dot{\theta}_L \\ \dot{\theta}_R \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = S^{-1}(q) \begin{bmatrix} \dot{x}_E \\ \dot{y}_E \\ \dot{x}_F \\ \dot{y}_F \end{bmatrix} \tag{2.30}$$

2.2.3 Dynamics of Mobile Platform

Assume that Equation (2.6) represents the dynamic model of the mobile manipulator in Figure 2.1. Expressing the first *m*-equations in Equation (2.6) yields:

$$M_{11}\ddot{q}_v + M_{12}\ddot{q}_r + C_{11}\dot{q}_v + C_{12}\dot{q}_r + F_v + A_v^T\lambda + \tau_{dv} = B_v\tau_v \tag{2.31}$$

Multiplying both sides of Equation (2.31) by S^T and using Equation (2.9) to eliminate the constraint force we obtain:

$$S^T M_{11}\ddot{q}_v + S^T M_{12}\ddot{q}_r + S^T C_{11}\dot{q}_v + S^T C_{12}\dot{q}_r + S^T F_v + S^T A_v^T\lambda + S^T\tau_{dv} = S^T B_v\tau_v \quad \text{or}$$

$$S^T M_{11}\ddot{q}_v + S^T M_{12}\ddot{q}_r + S^T C_{11}\dot{q}_v + S^T C_{12}\dot{q}_r + S^T F_v + S^T\tau_{dv} = S^T B_v\tau_v \tag{2.32}$$

Substituting both Equations (2.10) and (2.11) into Equation (2.32) yields

$$S^T M_{11} S\dot{v} + S^T M_{11} \dot{S} v + S^T M_{12} \ddot{q}_r + S^T C_{11} S v + S^T C_{12} \dot{q}_r + S^T F_v + S^T \tau_{dv} = S^T B_v \tau_v \quad (2.33)$$

To simplify the expression, Equation (2.33) can be rewritten as:

$$\bar{\tau}_v = \overline{M}_{11} \dot{v} + \overline{C}_{11} v + f_v + \bar{\tau}_{dv} \quad (2.34)$$

where

$$\overline{M}_{11} = S^T M_{11} S \quad (2.35)$$

$$\overline{C}_{11} = S^T C_{11} S + S^T M_{11} \dot{S} \quad (2.36)$$

$$\bar{\tau}_{dv} = S^T \tau_{dv} \quad (2.37)$$

$$\bar{\tau}_v = S^T B_v \tau_v \quad (2.38)$$

$$f_v = S^T (M_{12} \ddot{q}_r + C_{12} \dot{q}_r + F_v) \quad (2.39)$$

Component f_v consists of gravitational and friction force vector F_v, and the dynamics interaction with the mounted manipulator (i.e., $M_{12} \ddot{q}_r + C_{12} \dot{q}_r$). Let us consider the mobile platform as in Figure 2.1 (b). Assuming the dynamics interaction with the mounted manipulator is very small (the arm moves at low speed) and the platform moves in a horizontal *XY Cartesian* coordinate, the component fv can be neglected. Therefore, Equation (2.34) can be rewritten as:

$$\bar{\tau}_v = \overline{M}_{11} \dot{v} + \overline{C}_{11} v + \bar{\tau}_{dv} \quad (2.40)$$

or in the form of q generalized coordinates (Fukao *et al*., 2000):

$$\overline{B}(q)\tau = \overline{M}(q)\ddot{q} + \overline{V}(q, \dot{q})\dot{q} + \overline{Q} \quad (2.41)$$

where $\overline{Q}$ are the bounded unknown disturbance torques that could include the unstructured mobile platform dynamics. The matrices, $\overline{M}(q)$, $\overline{V}(q)$ and $\overline{B}(q)$ can be described respectively as:

$$\overline{M}(q)=\begin{bmatrix} \frac{r^2}{4b^2}(mb^2+I)+I_w & \frac{r^2}{4b^2}(mb^2-I) \\ \frac{r^2}{4b^2}(mb^2-I) & \frac{r^2}{4b^2}(mb^2+I)+I_w \end{bmatrix} \tag{2.42}$$

$$\overline{V}(q,\dot{q})=\begin{bmatrix} 0 & \frac{r^2}{2b}m_c d\dot{\varphi} \\ -\frac{r^2}{2b}m_c d\dot{\varphi} & 0 \end{bmatrix} \tag{2.43}$$

$$\overline{B}(q)=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.44}$$

where the mass of the total system, $m = m_c + 2m_w$ with m_c is the mass of the body, m_w is the mass of the wheel, $I = m_c d^2 + 2m_w b^2 + I_c + 2I_m$, and I_c is the moment inertia of body about the vertical axis thru *G*, I_w is the moment inertia of the wheel and motor about the wheel axis, and I_m is the moment inertia of the wheel and motor about the wheel diameter.

2.2.4 Dynamics of Manipulator Arm

Expressing the first *n*-equations in Equation (2.6), i.e. the equations related to the manipulator, yields:

$$M_{21}\ddot{q}_v + M_{22}\ddot{q}_r + C_{21}\dot{q}_v + C_{22}\dot{q}_r + F_r + 0 + \tau_{dr} = \tau_r \tag{2.45}$$

To clarify the dynamic interaction term, Equation (2.39) can be rearranged as:

$$M_{22}\ddot{q}_r + C_{22}\dot{q}_r + (M_{21}\ddot{q}_v + C_{21}\dot{q}_v + F_r) + \tau_{dr} = \tau_r \tag{2.46}$$

where the terms in the brackets, i.e. $(M_{21}\ddot{q}_v + C_{21}\dot{q}_v + F_r)$, are the dynamic interaction with the mobile platform and the gravitational and friction force vector.

For the manipulator with a planar configuration and moves in a horizontal *XY Cartesian* coordinate and the dynamic interaction is very small (the platform moves slowly), the component within the brackets in Equation (2.46) can be neglected so that it can be expressed as:

$$M_{22}\ddot{q}_r + C_{22}\dot{q}_r + \tau_{dr} = \tau_r \tag{2.47}$$

Equation (2.47) can be rewritten as:

$$\bar{H}(q)\ddot{q}_r + \bar{h}(q,\dot{q})\dot{q}_r + \tau_{dr} = \tau_r \tag{2.48}$$

where

$$\bar{H}(q) = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \tag{2.49}$$

and $$\bar{h}(q,\dot{q}) = \begin{bmatrix} 0 \\ m_2 l_1 l_{c2} \sin\theta_2 \end{bmatrix} \tag{2.50}$$

with $$H_{11} = m_2 l_{c1}^2 + I_{m1} + m_2\left(l_{c1}^2 + l_{c2}^2 + 2l_1 l_{c2}\cos\theta_2\right) + I_{m2} \tag{2.51}$$

$$H_{12} = H_{21} = m_2 l_1 l_{c2}\cos\theta_2 + m_2 l_{c2}^2 + I_{m2} \tag{2.52}$$

$$H_{22} = m_2 l_{c2}^2 + I_{m2} \tag{2.53}$$

2.2.5 Kinematic Control of Mobile Manipulator

There are a number of extensive works that can be found in literature related to the kinematic of the mobile manipulator (Wang and Kumar, 1993, Perrier *et al*., 1998, Bayle *et al*., 2001, Papadopoulos and Poulakakis, 2001, Sugar and Kumar, 2002, and Tanner *et al*., 2003). The works mostly focus on the methods to solve the redundancy problems with the dynamic not specifically addressed. The kinematic analysis is particularly useful to describe the robot's workspace and motion path

planning tasks including obstacles avoidance, collision free moving capability and manoeuvrability. Wang and Kumar (1993) used instantaneous (local) and global approaches of a screw-theoretic formulation to solve the redundancy of the robot motion. The basic of their works is the use of joints compliances analysis to resolve the redundancy. However the joint compliances model output equations resulted were not the *Jacobian* matrix of a kinematics function therefore it could not be directly applied to the nonholonomic system. The way they reformulated was by imposing additional constraints of the holonomic system so that the resulting system was holonomic. The screw-theoretical formulation approaches used in their works was found to be a theoretical significance although the approaches are relatively not the simple way to be chosen.

Kanayama, *et al*. (1990) proposed a stable tracking control method for an autonomous mobile robot. The major objective of the control rule is a method to find a reasonable target, linear and rotational velocities $(v,\omega)^T$. The stability of the control rule is proven using a *Lyapunov* function. According to Kanayama, one of the difficulties of the mobile robot problem lies in the fact that ordinary vehicles groups only two degrees of freedom (linear velocity and angular velocity) for locomotion control, although vehicles have three degrees of freedom, x, y and θ in its positioning. Another difficulty is in the non-linearity of the kinematic relation between $(v,\omega)^T$ and $(\dot{x},\dot{y},\dot{\theta})^T$. The use of a *Lyapunov* function also resolves these difficulties. Linearizing the system's differential equation is useful to decide parameters for critical dumping for a small disturbance. This equation is very popular in mobile robot literatures and has become a 'standard reference' by many researchers.

Perrier *et al*. (1998) implemented homogenous matrices and dual quaternion to represent the redundancies in the kinematics problems. They considered the global motion of mobile manipulator from point to point and computed a path that takes into account different constraints; the nonholonomic and holonomic. Their works were successfully resolved the redundancy of the kinematics problem in joints limitation, velocity and radius steering limitations although they didn't explain the

experimental results in their paper. Bayle *et al*. (2001) focused the investigation on the manipulability of a particular class of mobile manipulator in local kinematics analysis. They showed how the notion of manipulability could be extended to represent the operational methods in a configuration of the system. One of the advantages of their works is that it can be used to reconfigure the installed robot arms position on the top of the platform in order to maximize the workspace operations.

Papadopoulos and Poulakakis (2001) presented a robot trajectory tracking method with obstacle avoidance capability using conventional inverse kinematics control based on velocity control added by obstacle mapping formulation. In the paper they successfully showed the smooth effect and continuous function of trajectory tracking as well as polynomials characteristic and the simplicity in implementation of their scheme thru simulation. However, only the known obstacle location and fixed position can be considered because of the robot was not described to have self obstacle mapping algorithm. Sugar and Kumar (2002) developed a kinematics-based control of multiple mobile manipulators. Their analysis was based on tasks that require grasping, manipulation and transporting large and possibly flexible objects without special purpose fixtures. The kinematics problems were solved using the compliant arms analysis and it was successfully demonstrated in the paper for cooperation of two and three mobile manipulators. They also addressed an integrated force controller in their proposed scheme for each robot using force or stiffness control for the actively arms. They reported that the use of the force controller that affects the internal forces and moment on the box (the object handled) was successfully controlled, but the accuracy of the system was poor due to large orientation errors of the object.

Tanner *et al*. (2003) proposed a new methodology to motion planning for multiple mobile manipulators cooperation that applicable for articulated, non point nonholonomic robots with guaranteed collision avoidance and convergence properties. They implemented a potential field technique using *diffeomorphic* transformations and the resulting point-world topology. Their approach was applied on multiple mobile manipulators to handle deformable material in an obstacle environment and it showed the success thru simulation. The main feature of their

method is application of dipolar potential field technique to guarantee the robot will approach the destination asymptotically and it will follow the path that automatically stabilizes its orientation. This method incorporated an inverse *Lyapunov* function to stabilize (and converge) the robots navigation. Their works are found very useful for future developments in coordinated kinematics controls, but unfortunately they did not address the robustness of each robot specifically.

2.2.6 Dynamic Control of Mobile Manipulator

In actual implementation pertaining to robot's motion, it is also typical to address the control problem involving the robot's dynamic. At this juncture, designing appropriate controller can lead to significant improvement in performances (Yamamoto and Yun, 1996). Combining both the extensive kinematic and dynamic aspects for a perfectly motion control of any dynamical system still remains a complex and challenging problem. In recent years, several researchers have contributed to solving this problem using a number of methods such as those related to adaptive (Colbaugh, 1998), optimal (Mohri *et al*., 2001), neural network (Lin and Goldenberg, 2001), and model-based (Papadopoulos and Poulokakis, 2000) control schemes. Yamamoto and Yun (1996) rigorously modeled the dynamics coupled and interaction between mobile platform and manipulator. They have investigated the dynamics interactions when the manipulator executing the task and the platform has followed a specified trajectory. The aim of their work was how to actively compensate the dynamic interaction that influences to the robot performance using a class of non-linear feedback control. The feedback control was based on state space model. In the paper, they showed the successful implementation of the technique to mobile manipulator by way of simulation.

Colbaugh (1998) addressed the problem of stabilizing mobile manipulators in the presence of uncertainties regarding the system dynamic model. He proposed an effective solution by combining homogenous system theory and adaptive control theory. His proposed scheme contains of two subsystems, i.e. a homogenous kinematic stabilization strategy which derived a desired velocity based trajectory,

and an adaptive control scheme which ensures that the velocity based trajectory is accurately tracked. Mohri *et al*. (2001) presented a trajectory planning method using an optimal control theory. They derived the robot's dynamics by considering it as a combined system of mobile platform and manipulator. Then they applied a sub-optimal trajectory planning using an iterative algorithm based on gradient functions synthesized in the hierarchical manner to formulate the trajectory-planning problem. They reported the effectiveness of the scheme by simulation. One of the advantages is the simple trajectory planning modification needed when an obstacle is considered in the workspace. Nevertheless they did not state the overall system robustness specifically in the paper.

Lin and Goldenberg (2001) proposed a class of neural network (NN) control of mobile manipulator that is subjected to kinematics constraint. They assumed that the robot's dynamic is completely unknown and it would be identified using a neural network estimator. For the trajectory tracking control they used a class of feedback control that its stability was tested using a *Lyapunov* function theory. They have used a procedure to estimate the dynamics as follows: First, the robots dynamics were redefined as an error dynamics based on a set of carefully chosen *Lyapunov* sub-functions through a joint-space tracking. Next, a NN on-line estimator was constructed and a new-NN learning law was obtained after which a new NN control could be derived. From the simulation, they were able to demonstrate the effectiveness of their proposed scheme.

Papadopoulos and Poulokakis (2000) used a model-based approach to derive the control of mobile manipulator. The robot dynamics was assumed to be completely known. They did not use any specific robust control scheme incorporated into the motion control except the error-based feedback control that consider the dynamics coupling of the mobile platform and the manipulator. The robustness of the system was not specifically guaranteed because they only applied the position and velocity orientation.

2.3 Resolved Motion Rate Control and Resolved Acceleration Control

Resolved Motion Rate Control (RMRC) in robotic means that the motions of the various joint or degrees of motors are combined and resolved into separately controllable motions along the world coordinate axes (Fu *et al*., 1987). This implies that the motors must run simultaneously at different time-varying rates in order to achieve the coordinated motion in the workspace. The parameters controlled could be parts of the order of the control system, i.e. position, velocity and or acceleration. In feedback control system, the resolved motion control can be grouped in two types, i.e. resolved motion rate control (RMRC) and resolved motion acceleration control (RMAC). The second is often simply described as resolved acceleration control (RAC). The main difference of the methods is that the RMRC employs position and velocity orientation as the controlled parameters only while the RAC also incorporates acceleration signals in order to achieve a more stable control.

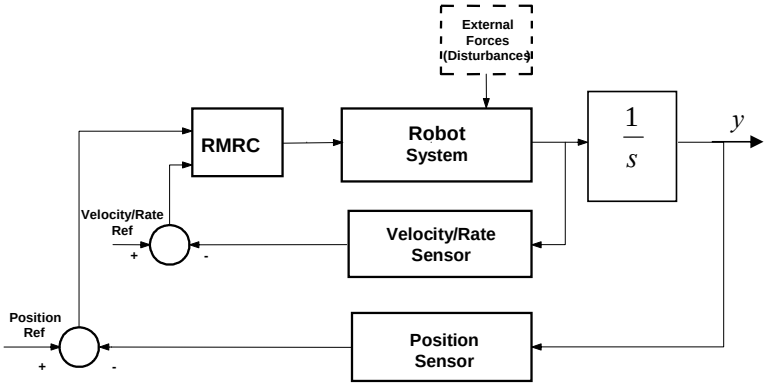


Figure 2.2: A schematic diagram of RMRC

A schematic diagram of RMRC is illustrated in Figure 2.2. The figure depicts the input functions consisting of velocity/rate and position. This implies the users can specify any position and rate of each motor/actuator independently. In contrast, a simultaneous velocity and position calculation should be done carefully. Nevertheless, the RMRC is still one of the simplest motion controls that is selected by a number of researchers in the last decade including RMRC using fuzzy logic (Kim and Lee, 1993), singularity free RMRC (O'neil *et al*., 1997) and inverse

kinematic learning for RMRC applied to a humanoid robot (D'souza *et al*., 2001). Luh *et al*. (1980) introduced an extension to the RMRC by incorporating acceleration command vectors in the schematics. The scheme was popularly known as resolved motion acceleration (RMAC) or simply resolved acceleration control (RAC). The general form of the RAC in robotics can be illustrated as shown in Figure 2.3.

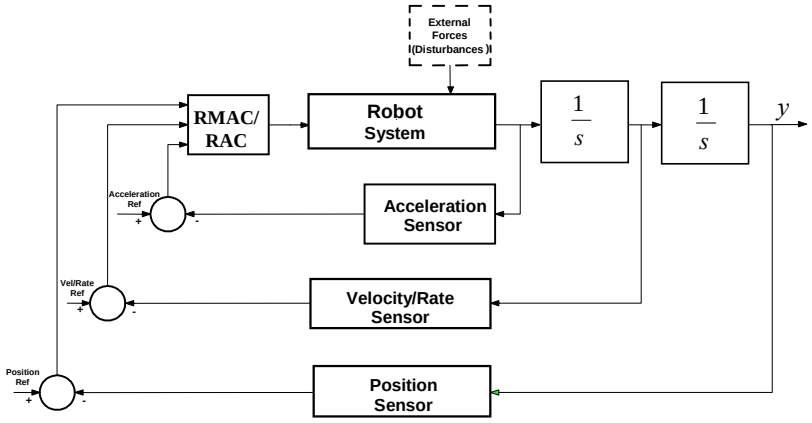


Figure 2.3: An illustration of the RMAC/RAC schematic diagram

Muir and Neumann (1990) investigated the use of RMRC and RAC in a mobile robot control. They were able to apply the RAC scheme in a mobile robot system and it was demonstrated that RAC was superior compared to RMRC. Kircanski and Kircanski (1998) reported a rigorous investigation of RMRC and RAC applied to a damping-based braking system with actuators constraints. They have showed that the addition of braking system to the RAC was found to be more effective than before. Campa *et al*. (2001) proposed a method to prove the local asymptotic stability of RAC applied to robot manipulator control by invoking a strict *Lyapunov* function. They concluded that the motion control of the manipulator in task space and whose pose of the end-effector is of some concern, could be solved by the RAC technique.

From the above reports, it is obvious that the RAC is still one of the best control options due to its simplicity in real-time implementation. In the study, the RAC is developed as the integrated simplified mobile platform coordinate and

heading angle, (x_v, y_v, φ) control and the *XY Cartesian* planar manipulator's tip position coordinate, (x_m, y_m) control. By using this technique, the proposed control scheme would have a more flexible position, speed and acceleration control. This flexibility is gained by the use of simultaneous input reference position, velocity and acceleration parameters.

2.4 Active Force Control (AFC)

Hewit and Burdess (1981) first applied the technique known as active force control (AFC) successfully to a robotic arm. They proposed a practical and robust technique to compensate for the internal and external disturbances of a mechatronics/machinery system by employing an internal force error feedback control based on real-time acceleration and force measurements. The general form of the AFC scheme is depicted in Figure 2.4.

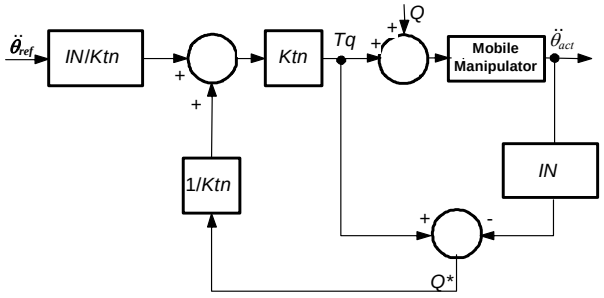


Figure 2.4: Typical AFC loop

The important AFC algorithm is defined as follows:

$$Q^* = T_q - \mathbf{IN}\ddot{\theta}_{act} \tag{2.56}$$

where $\ddot{\theta}_{act}$ is the actual acceleration signal, T_q is the torque applied, Q is known and unknown (bounded) internal and external disturbances including payloads, Q^* is the

estimated disturbances observed and **IN** is the estimated inertia matrix, approximated by suitable means. In the figure, $\ddot{\theta}_{ref}$ is the reference acceleration signal and K_{tn} is the motor torque constant. The aim of the AFC method is to ensure that the system is stable and robust even in the presence of known or unknown disturbances. The main advantage of the method is that it can give a compensating action using mainly the estimated or measured values of certain parameters. This method has the benefits of reducing the mathematical complexity of the robot system that is known to be highly coupled and non-linear. However, AFC could not solve the motion control of robotic system by itself without incorporating a lower order of control, i.e., position and or velocity control, typically located in the outermost loop. The scheme depicted in Figure 2.4 is an open loop mode of control so that it should be combined with a proper feedback (tracking) control in order to achieve a complete motion control. One of the alternatives is by implementing the RAC scheme to the AFC which is one of the proposed schemes implemented in the research. Figure 2.5 shows the concept of a RAC combined with AFC that is applied to a robot system. In the figure, the RAC is depicted to contain in the left dashed box. Input references in the RAC consist of position (x_{bar}), velocity (x_{dbar}) and acceleration (x_{ddbar}) that are all subjected to the workspace of the coordinate system. The right dashed box contained in the figure indicates the AFC part. **H** denotes the dynamic constraint of the robot system. The main computational burden in AFC is the multiplication of the estimated inertia matrix with the respective acceleration of the robot dynamic component before being fed into the AFC feed forward loop.

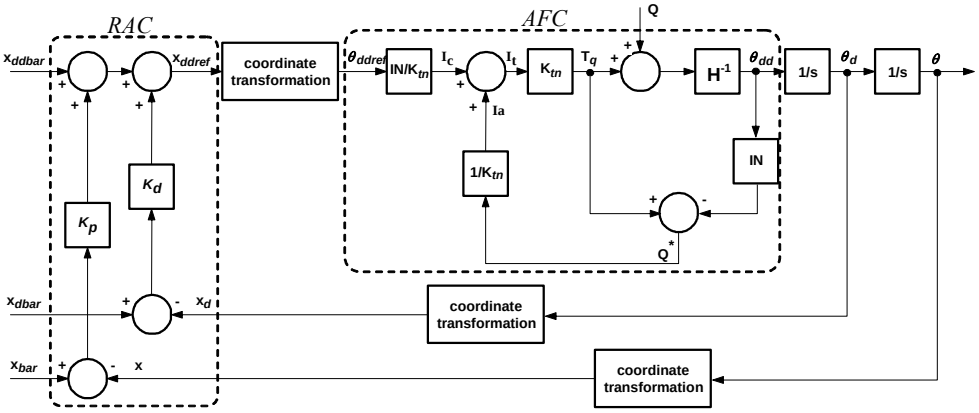


Figure 2.5: RAC combined with AFC

A number of implementations of the AFC schemes applied to robotic manipulator can be found in Hewit and Marouf (1996), Musa (1998), Hewit and Morris (1999) and Pitowarno *et al*. (2001). In other forms, the basic AFC scheme was further studied and modified by a number of researchers and can be found in works by Uchiyama (1989), Ohnishi (1995), Komada *et al*. (1996), and Godler *et al*. (2002). The term 'disturbance observer control' as a class of the modified AFC was typically used in literature. Some of their proposed modified AFC schemes were successfully implemented in real world applications.

2.4.1 Estimated Inertia Matrix

As mentioned earlier, one of the main problems in AFC scheme is how to acquire the appropriate estimated inertia matrix so that the compensation of the disturbances can be effectively carried out. Actually, the inertia matrix of the robotic system when operated is highly non-linear that can be caused by payloads, noise, frictions such as static, coulomb, and viscous friction, vibrations and other unknown disturbances. Supposed that the system has relative small disturbances (inherent noise at sensors and or actuators, low friction at joint and no/small vibration), so the appropriate inertia matrix can be sought using basic approximation techniques. Basically, the inertia matrix of a specific robotic system - for example two-link planar robotic arm – can be heuristically introduced first with a certain value by considering the system and mass properties.

The application of AFC to a planar two-link robotic arm using the basic (crude) approximation technique was successfully presented (Musa, 1998). The tested robot system has the estimated inertia matrix expressed in the following form:

$$\mathbf{IN} = \begin{bmatrix} IN_{11} & IN_{12} \\ IN_{21} & IN_{22} \end{bmatrix} \tag{2.55}$$

and $$\mathbf{IN} = K_i \, \mathbf{H} \tag{2.56}$$

where the off-diagonal elements $IN_{12} = IN_{21} = 0$.

Although the most appropriate inertia matrix estimator should indicate properties of “non-linearity” of the value following the high non-linearity of the system operation – not a fixed value, K_i, as in Equation (2.56) – however this basic multiplier constant as an estimator was sufficiently exposed the powerful of the AFC in the application (Musa, 1998).

2.4.2 Estimation of Inertia Matrix using Intelligent Schemes

Some estimation of the inertia matrix techniques applied to robotic systems, mostly to robotic arm systems, were successfully implemented, such as using *Acceleration Tracing Orientation Method* (ATOM) (Komada and Ohnishi, 1990), disturbance observer and inertia identifier in task space (Komada, *et al*., 1996). In particular, also in a robotic arm system, the robustness of control schemes with disturbance observer and with acceleration control loop was compared and reported (Godler *et al*., 1999). So far there is no report related to applications of the inertia matrix estimation technique of the AFC that is implemented to a mobile manipulator system. In the case of planar two-link robotic arm, some implementations of intelligent techniques for estimating the inertia matrix were presented, such as using neural network (Musa, 1998), fuzzy logics (Musa and Nurul, 2000), iterative learning (Musa and Hooi, 2000), and knowledge-based method (Pitowarno, 2002). The rationale of using the intelligent mechanism is to compute the required inertia matrix automatically, continuously and on-line. However, the using of these intelligent techniques in the AFC was merely dealt with the robotic arm only. In the research work the estimation of the inertia matrix on a more complicated robotic structure, i.e. the mobile manipulator system would be intensively investigated. In the subsequent chapters a class of iterative learning and knowledge-based fuzzy algorithms as the inertia matrix estimator of the robot which is proposed in this study would be relevantly described in the ensuing sections related.

2.5 Knowledge Based System

The term of knowledge-based system (KBS) is commonly used to represent a system that work in case of handling all of the system aspect or operation that based on a group of prior knowledge of the system's characteristics, behavior, environments and or all others information – or in the close term: knowledge - that can be correlated to the system. In other point of view the KBS is a system with mostly human-centered. If the human have expertise to interpret the knowledge to a smart knowledge-based decision rules and this expertise to be represented to a computer then the overall system may be described as an expert system.

Generally, the major component of a knowledge-based system is well known as expert system. As mentioned previously, expert system is a term used to represent an intelligent technique that is mainly based on the human expertise (the knowledge of the expert) whether directly or indirectly. Soucek (1989) described some typical engineering applications of expert systems in fault diagnosis, factory cell simulation, knowledge-base robot and process control. These expert systems follow the concept of *Model-Based Reasoning* that the model contains an explicit description of a domain, objects, relationship between objects, and taxonomy of the object classes. By the approaches, the expertise could be interpreted as rules expression that will be used in the computer or controller programming.

Figure 2.6 shows an illustration of a generic expert system as mentioned by Ignizio (1991). The components within the dashed box are those linked with the use of computer or microprocessor. Outside this dashed box, access capabilities for two types of human users are noted. The first one is the individual designated as the knowledge engineer that is the person responsible for placing the knowledge into the expert system's knowledge base. He or she accomplishes this job through the *interface* - as shown as directly interconnected, and the *rule adjuster*. The knowledge engineer can also be the interface between the human expert and the expert system. This means, the knowledge engineer somehow must capture the expertise of the human expert and express it in a format or structure that may be stored in the knowledge base memory that will be used by the expert system.

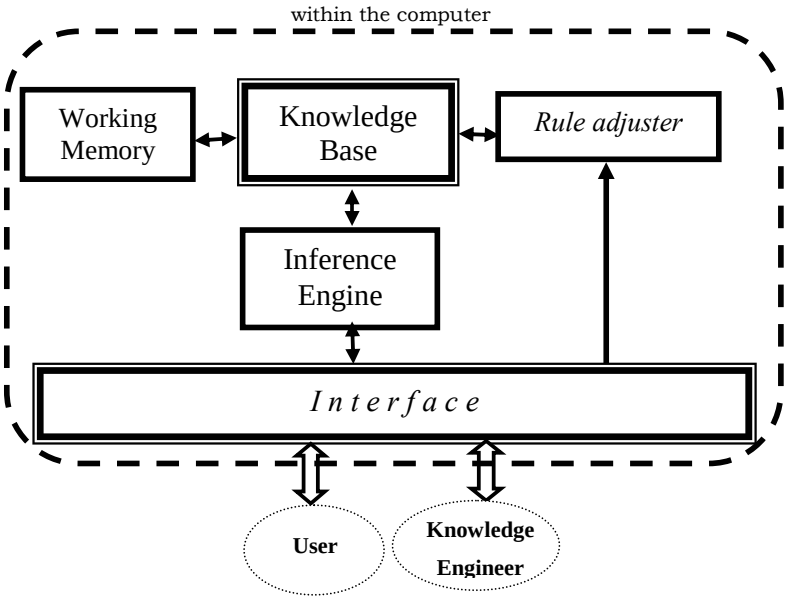


Figure 2.6: An illustration of a generic expert system

Ideally, there would be no need for a knowledge engineer by assumption that the domain expert would directly interact with the expert system. So the domain expert would replace the knowledge engineer in the figure. The second type of individual with access to the expert system is the *user*. This designation refers to anyone who will use the expert system as a decision-making aid.

The *interface* handles all input to or output from the computer. Currently, the interface becomes more sophisticated in line with advancement of computers' technology. The input and output devices are already improved to meet the demands of the multimedia and global networking era. The *inference engine* performs two primary tasks. Firstly, it examines the status of the knowledge base and working memory so as to determine what facts are known at any given time, and to add any new facts that become available. Secondly, it controls the session by determining the order which inferences are made. The *knowledge base*, as the heart of the expert system, contains two types of knowledge, i.e., facts and rules. The *working memory* is a memory device that the contents change according to the specific problem at hand. Finally, the *rule adjuster* serves merely as a rule editor that enters the rules specified by the knowledge engineer into the knowledge base during the

development phase of the expert system. In more sophisticated expert systems, the rule adjuster may be improved by incorporating a learning algorithm in the process. In particular, Young-Tack and Wilkins (1992) depicted that the KBS inference mechanism could be an intersection of expert system (based on AI or Intelligent Control methods), machine learning and probabilistic reasoning. Figure 2.7 illustrates this concept.

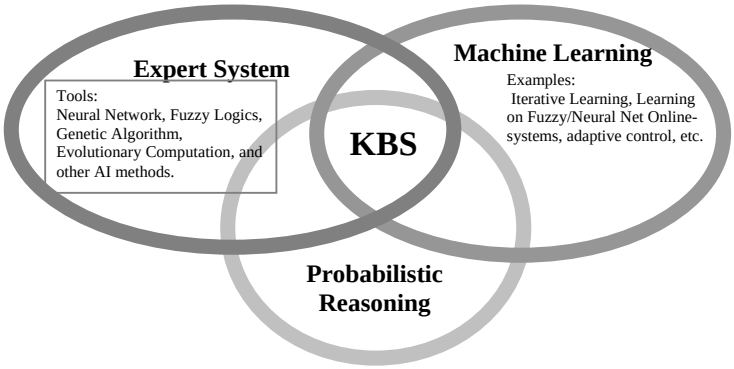


Figure 2.7: An illustration of KBS

The explanation using Figure 2.7 gives more precise meaning of KBS inference mechanism. Each of the three parts of KBS could have particular domain in research. Currently, designing an appropriate knowledge-based method to a system not only needs expert system methodologies to realize, but however machine learning algorithms are also needed to create knowledge includes rule adjuster automatically. On the other hand, to ensure the validity of knowledge being collected statistical based algorithms are also important to be influenced. Machine learning is the study of computer algorithms that improve automatically through experience (Mitchell, 1997). Applications range from data mining programs that discover general rules in large data sets, to information filtering systems that automatically learn user's interests. Machine learning methods have been applied to problems such as learning to drive an autonomous vehicle, learning to detect credit card fraud, learning to recognize human speech, etc. The probabilistic reasoning as the third group of the intersection, also gains through the exploration of the goal of KBS.

Recently, in common application the KBS are designated to handle a large amount of knowledge with complex interconnections that need a probabilistic analysis to manage it. Thus, the KBS environment also needs the statistical approaches to solve this problem. Some researchers focused the research on the probabilistic analysis of KBS. Growe *et al*. (2000) described in detail how the *Bayesian Network* could be implemented as a powerful tool to judge validity of the knowledge base. In particular, the human expert supplies the observations of the probabilities in any possible way. So, the accuracy of probabilities value (0 to 1) is fully dependent on the statistical based observation of the data or knowledge being collected. The human expert can use the analytical based and or survey based to set this value. Hence, the accuracy of the analysis also depends on how valid the probabilities being set.

Generally, the KBS procedures or mechanism consisted of *knowledge investigation and validation*, *knowledge representation*, *knowledge acquisition* and *knowledge processing*. From these procedures, the inference mechanism can be developed and implemented by way of a computer program. In the last decade (1995 to 2004) expert system methodologies were rigorously investigated by researchers. This implies the use of some terms in expert system domain, KBS and other human-centered artificial intelligence (AI) method could be interchangeable. A survey that indicated this development in expert system methodologies was reported by Liao (2005). He has found that some existing terms often used by researchers to describe the specific meaning in their expert system applications, such as rule-based system, knowledge-based system, neural networks, fuzzy expert system, object-oriented methodology, case-based reasoning, modeling and system architecture, intelligent agents, ontology, and database methodology. It is usual that some of the terms like ‘rule-based’ and ‘knowledge-based’, ‘rule-based and fuzzy logic’ are interchangeable and could be overlapping.

2.6 Fuzzy Logic

In general, a fuzzy logic (FL) concept as initiated by Zadeh (1988) attempts to mimic human experts' knowledge. It tries to imitate the human expert's behavior by computing with words. First, human words are depicted by membership of functions (MFs). Secondly, qualitative human knowledge is contained in the rule base. Then, experienced human reasoning is imitated by implication methods and the inference engine. For the purpose of reviewing some principles of the equations used in designing a fuzzy logic system, they will be explained briefly in the following paragraphs.

Figure 2.8 shows a block diagram of a typical fuzzy logic system.

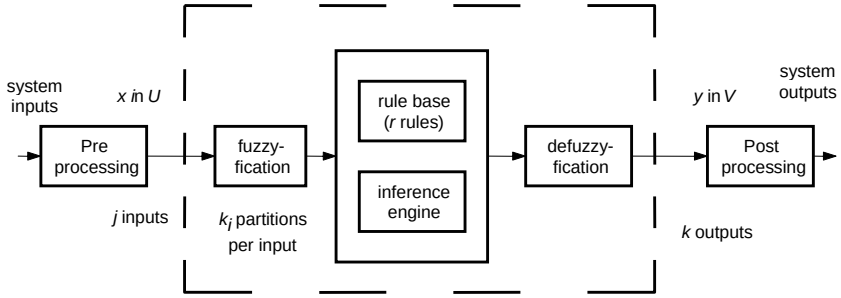


Figure 2.8: An illustration of a fuzzy logic system

The input space fuzzifier or MF has shape and parameters to design. The shape at least consist of singleton, trapezoidal, *Gaussian* MF, while the parameters provide support, crossover, and bandwidth. The input space partition at least has options of uniformity and completeness. Users can choose these options of MF depending on their needs and requirements. The rule base has typically two options, i.e. *fuzzy antecedent – fuzzy consequence* and *fuzzy antecedent – crisp consequence* (singleton or linear function). The equations of singleton, trapezoidal and *Gaussian* MFs are given in Equations (2.57), (2.58) and (2.59) respectively.

$$\mu(x,a,b,c)=\begin{cases}0 & x<a\\ \dfrac{x-a}{b-a} & a\le x\le b\\ \dfrac{x-c}{b-c} & b\le x\le c\\ 0 & c<x\end{cases} \tag{2.57}$$

$$\mu(x,a,b,c,d)=\begin{cases}0 & x<a\\ \dfrac{x-a}{b-a} & a\le x\le b\\ 1 & b<x<c\\ \dfrac{x-d}{c-d} & c\le x\le d\\ 0 & d<x\end{cases} \tag{2.58}$$

$$\mu(x,p,w)=e^{-\left(\frac{x-p}{w}\right)^2\ln(2)} \tag{2.59}$$

The rule base with r rules is given by:

$$r=\prod_{i=1}^{j} m_i\ , \tag{2.60}$$

can be described as follows:

$$R^z:\textbf{IF}\underbrace{<\textit{proposition}>}_{\textit{antecedent}}\textbf{THEN}\underbrace{<\textit{proposition}>}_{\textit{consequence}} \tag{2.61}$$

where z is rule number, $1\le z\le r$.

For option of *fuzzy antecedent – fuzzy consequence* the rule can be expressed as follows:

$$R^z:\text{IF x1 is } A_1^z \text{ AND } \ldots x_k \text{ is } A_k^z \text{ THEN } y \text{ is } B_z \tag{2.62}$$

For *fuzzy antecedent – crisp consequence* the equation is:

$$R^z : \text{IF x1 is } A_1^z \text{ AND } \ldots x_k \text{ is } A_k^z \text{ THEN } y = \eta_z^T x + c_z, \qquad (2.63)$$
$$\eta_z^T = [\eta_{z1}, \eta_{z2}, \ldots, \eta_{zj}]$$

In the fuzzy system each rule mentioned above is basically evaluated using *Generalized Modus Ponens* (GMP) inference, i.e.:

Premise	:	IF x is A THEN y is B
Antecedent	:	x is A'
Consequence	:	y is B'

Commonly, this evaluation result, β_z, can be implicated in two options, i.e.:

$$\beta_z = \prod_{i=1}^{j} \mu_{A_j}^z(x_i) = \mu_{A_1}^z(x_1) \text{ and } \mu_{A_2}^z(x_2) \text{ and } \cdots \mu_{A_j}^z(x_j) \qquad (2.64)$$

or

$$\beta_z = \min_i(\mu_{A_j}^z(x_i)) = \mu_{A_1}^z(x_1) \text{ and } \mu_{A_2}^z(x_2) \text{ and } \cdots \mu_{A_j}^z(x_j) \qquad (2.65)$$

where Equation (2.64) is often called *product implication*, while Equation (2.65) is *minimum implication*. In design, the rule's consequences aggregation can also be defined in two ways, i.e. *weighted* and *maximum* as described as follows:

$$\mu_{B_1'}(y) \text{ or } \mu_{B_2'}(y) \text{ or } \cdots \mu_{B_r'}(y) = \sum_{i=1}^{r} \mu_{B_i'}(y) = \mu_{B'}(y) \qquad (2.66)$$
$$\forall y \in V$$

or

$$\mu_{B_1'}(y) \text{ or } \mu_{B_2'}(y) \text{ or } \cdots \mu_{B_r'}(y) = \max_i(\mu_{B_i'}(y)) = \mu_{B'}(y) \qquad (2.67)$$
$$\forall y \in V$$

where $\mu_{B_z'}(y)$ as the inference engine operator can be selected from type of *Mamdani* fuzzy consequence or *Larsen* fuzzy consequence (Olivares, 2003), i.e.:

$$\mu_{B'_z}(y) = \min(\mu_{B_z}(y), \beta_z) \qquad \forall y \in V \tag{2.68}$$

or

$$\mu_{B'_z}(y) = \mu_{B_z}(y)\beta_z \qquad \forall y \in V \tag{2.69}$$

Finally, this section also presents a brief review of the last step of operation in a fuzzy system, i.e. defuzzification. It has two options, i.e. *centre of gravity* and *mean of maxima*. The center of gravity has mathematical expression as follows,

$$y_0 = \frac{\int_{y \in V} \mu_{B'}(y) y dy}{\int_{y \in V} \mu_{B'}(y) dy} \tag{2.70}$$

while the mean of maxima is:

$$y_0 = \frac{\int_{hgt(B')} y dy}{\int_{hgt(B')} dy} \tag{2.71}$$

where

$$hgt(B') = \arg\max_{y \in V}(\mu_{B'}(y)) . \tag{2.72}$$

For direct crisp output of crisp rule's consequences aggregation the weighted average y_0 can be described as:

$$y_0 = \frac{\sum_{i=1}^{r} \beta_i (\eta_i^T x_0 + c_i)}{\sum_{i=1}^{r} \beta_i} \tag{2.73}$$

From the options described in Equations (2.57) to (2.73) the most used fuzzy systems in system engineering and control, namely the *Mamdani* type and *Takagi-Sugeno* type can be characterized as follows: *Mamdani* fuzzy system has minimum implication, membership functions as consequence, maximum aggregation, and

centre of gravity defuzzifier, while the *Takagi-Sugeno* type has product implication, linear input function as consequence, and weighted average aggregation defuzzifier. This section shall adequately presents a brief review of fuzzy systems. However, the quality of a fuzzy controller design does not merely depend on the types or options selected, but it is implied by how the controller should be properly set to meet with the system being controlled.

2.7 Knowledge Based Fuzzy Control

The term 'knowledge-based fuzzy control' was clearly first quoted in Rhee *et al*. (1990). The term consists of two components, i.e. knowledge-based system (KBS) and fuzzy control. These two parts can be deducted as systems or methods that employ expertise of human to derive its explanations, procedures, algorithms and its concluded rules that will be applied. In other descriptions, systems that the controls are influenced with human expertise can be considered as expert system. Expert system is a term used to represent a 'method of control' that mainly based on the human expertise (the knowledge of the expert) whether directly or indirectly. Directly means the knowledge of the expert is supplied to the system online, and indirectly means the knowledge is reprocessed as rules.

Ignizio (1991) defined expert system as a computer program (or methods of computation) that exhibits, within a specific domain, a degree of expertise in problem solving that is comparable to that of human expert. These expert systems follow the concept of *Model-Based Reasoning* that the model contains an explicit description of a domain, objects, relationship between objects, and taxonomy of the object classes, Soucek (1989). In contrast, fuzzy logic control that initiated by Zadeh basically has inherent problems on reasoning in designing rules. Ordinary fuzzy control has no adaptation algorithm to reorganize the rules when system is running in which the common way to design the proper rules are usually heuristically approach. To solve the problem, several researchers intensively investigated techniques in order how to make fuzzy control online and having adaptation capability, such as Jang

(1993), Mar and Lin (2001), and Hassanzadeh *et al*. (2002). If the online adaptive fuzzy control is not a simple way to be implemented in certain control system case, optimizing the fuzzy rules thru knowledge-based reasoning can be a considerably choice.

Rhee *et al*. (1990) rigorously described how to implement a knowledge-based fuzzy logic in control systems. They investigated a method for fuzzy control of system based on a concept of human thinking. Their fuzzy controller consists of a knowledge-based containing a number of process time responses. The basic elements employed were cells in which an input and corresponding output of system were stored. The cells were organized according to different types of relations that the result was defined as knowledge structure. They distinguished the control design in two phases. The former is learning process (knowledge investigation and representation) in which the relations between input and corresponding output are investigated. In this phase the knowledge structure is constructed. Then, the latter phase is implementation of the knowledge structure to control the process. In the research project a similar way of knowledge-based fuzzy (KBF) control incorporated to AFC scheme is proposed in order to design a robust motion control of mobile manipulator. The application of the knowledge-based fuzzy logic mentioned above to feedback controls is rarely found in literatures. Most of the implementations are in data retrieval and image segmentation/processing that is very useful in medical diagnosis system.

An investigation on knowledge-based fuzzy logic method that used genetic algorithm to refine the knowledge representation and acquisition can be seen in Ouchi and Tazaki (1998). They used a fuzzy petri-net to perform fuzzy reasoning. However their method was applied to medical diagnosis system and it should be combined with a feedback control to perform the use in engineering field such like robotics. Nanayakkara and Samarabandu, (2003) implemented a class of knowledge-based fuzzy logic to automatic model-based image segmentation system in medical diagnosis system. They successfully showed the implementation on ultrasound image of prostate.

2.7.1 Knowledge Based Reasoning in Fuzzy System

As mentioned previously, a fuzzy system attempts to mimic human experts' knowledge. This means the building of a fuzzy system can be considered as a fuzzy expert system. A good fuzzy system should include all of the human expertise aspects into the design. However, in a procedural fuzzy system design there is still classical problems in reasoning. Theoretically there is no specific reasoning guidance to design, for example, what are the best of membership of functions and its configuration. Basically, reasoning in fuzzy system or fuzzy expert system design can be achieved through two ways, i.e. *forward chaining* and *backward chaining* inference engine (Negnevitsky, 2002). This follows the basic reasoning of rule-base expert systems as well as for fuzzy expert systems. If an expert (human) first needs to gather some information and then tries to infer from it whatever can be inferred, the forward chaining inference engine is the choice. In contrast, if the expert begins with a hypothetical solution and then attempts to find facts to prove it, the backward chaining inference engine is the suitable choice. One of the proposed methods in this study, i.e. *Knowledge Based Fuzzy Active Force Control*, uses a method that similar with the backward chaining inference engine in the design. A typical process in developing the fuzzy expert system with backward chaining inference engine approach (Negnevitsky, 2002), incorporates the following steps:

a) Specify the problem and followed by defining linguistic variables.
b) Determine fuzzy sets.
c) Elicit and construct fuzzy rules.
d) Encode the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference.
e) Evaluate and tune the system.

In a backward chaining approach the steps mentioned above can be modified as follows:

a) Specify the problem.
b) Investigated the relation of the input and output functions.

c) Defining linguistic variables.
d) Elicit and construct fuzzy rules.
e) Encode the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference.
f) Evaluate and tune the system.

A latest example of an application of knowledge-based fuzzy control can be found in Ciliz (2005). His contribution is about tuning algorithm and rule-base reduction through knowledge-based investigation. He proposed an algorithm that computed the inconsistencies and redundancies in the overall rule set based on newly proposed measure of equality of the individual fuzzy sets. He reported that the algorithm was successfully tested experimentally for the control of a commercial household vacuum cleaner.

2.8 Motion Planning

In past two decades, the motion planning problems have been extensively studied and it is in fact one of the better-formulated problems in robotic study. Generally motion planning transforms the high-level mobile robot's commands into a sequence of low-level motor actuating signals. Through motion planning, the robot user is able to specify the desired high-level activity with the system itself that fills in the missing low-level details (Sharir, 1989). Motion planning of the mobile robot involves various subfields and disciplines, such as path planning and trajectory planning.

2.8.1 Path Planning

One of the main aspects in motion planning is the path planning which addresses the geometric concerns of robot motion without regard to time. There are two stages involved in path planning, i.e global path planning and local path

planning. For global path planning, the robot system's geometry is fixed to either a circle or polygon, while the environment can be arbitrary. However, for local path planning, instead of fixing the robot system's geometry, the operating environment of the robot is fixed. In this project, only global path planning is considered. Generally, path planning can be solved by two approaches, they are model-based path planning and sensor-based path planning.

2.8.2 Trajectory Planning

Trajectory planning concerns with the time history of a series of the collision-free configurations which are obtained through path planning (Aydin and Temeltas, 2002). In this case, trajectory planning is actually an interface between the motion planner with the motion controller. Trajectory planning obtains the time-independent geometry points from the path planner and at the same time instills the time element into the planned path to transform it into the time-indexed trajectory. Several approaches have been introduced in past few years for the formulation of the trajectory planner.

Zulli *et al.* (1995) had proposed an interface between the motion planner with the motion controller. The interface transformed the output of the motion planner which was presented as a sequence of arcs and segments, into a time-indexed trajectory. Later, Hu *et al.* (1997) introduced the hybrid control architecture for the mobile robot motion planning. An intelligent expert system was constructed for the selection of the best motion command from the motion planner which was then fed into the motion controller.

Piazzi and Guarino (2000) proposed the *Quintic G^2-splines* for the trajectory planning of the mobile robot. *Quintic G^2-spline* interpolates an arbitrary sequence of points with overall second order geometric continuity.

2.9 Virtual Reality (VR)

With the advent of the first model drawing, design, and construction by the aids of the computer, the next logical step will be the use of virtual reality (VR). VR is based on a model of reality where the computerized information is represented as a separate, two or three-dimensional object in the computer and it can be experienced directly and interactively by the participant to provide him a sense that he is inside the data space (Schmitt, 1993). Broadly, there are three forms of VR, typically referred to as *desktop VR*, *projection VR*, and *immersive VR*.

Desktop VR resorts to a range of computing equipments, from personal computers to powerful graphics workstation in order to enable the participant to build and interact with the virtual worlds. Meanwhile, *projection VR* differs from *desktop VR* in the sense that it polarizes the computerized graphical images from a pair of conventional video projectors to provide groups of individuals a large-screen stereoscopic graphic (Stone, 1995). *Immersive VR* is more to the description of the participant's experience of "being there" in the virtual environment through VR interaction tools, such as the head-mounted displays and interactive gloves. In this project, the developed VR wheeled mobile robot simulator is based on the concept of *desktop VR*.

The principles of VR are *immersion* and *interactivity*. *Immersion* is achieved through blocking out all sorts of distractions from the participant so that he can focus selectively on just the information which he wants to work with. *Interactivity* means the ability for the participant to interact with the events in the virtual world (Zheng, *et al.*, 1998). Generally, a VR system should consist of three aspects that should be taken into account. It should be able to response to participants or users actions, provide real-time 3D graphics and instill a sense of immersion into the participants vision. Since VR is still considered by many as relatively an embryonic technology, it's potentials are largely open-ended.

2.10 Conclusion

A number of theories, fundamental concepts and relevant literature related to the research have been highlighted in this chapter. These shall serve as the basis to design a number of robust AFC-based systems effectively since the AFC part relies heavily on the appropriate technique to estimate the real-time inertia matrix in the AFC feed forward control loop of the mobile manipulator. From the study, it is feasible to improve the robot control scheme by implementing an intelligent method like fuzzy logic or knowledge-based system due to the complexity and the non-linear aspects of the system and its environment. A deeper understanding of the mobile robot's interaction with the environment can be explored by means of a virtual reality (VR) technique embedded with the motion planning and control. All the essential elements described in this chapter shall be suitably employed in the proposed control schemes which will be duly described in succeeding chapters.

CHAPTER 3

ESTIMATION OF INERTIA MATRIX BASED ON $v\omega$-ACTIVE FORCE CONTROL WITH FUZZY LOGIC

3.1 Introduction

In this chapter, an inertia matrix estimation method using fuzzy logic (FL) for an AFC scheme is presented. Generally, the inertia matrix of a mobile robot is not fixed, but it is changing with the motions of the mobile robot. Therefore, in relation to the changes of the mobile robot motions, the estimated inertia matrix **IN** should be approximately adapted for maximum performance of the mobile robot. The aim of this chapter is to improve the inertia matrix estimation method by switching the usage of crude approximation method to the proposed FL method. The FL mechanism computes **IN** automatically and continuously, while the mobile robot is tracking the reference trajectory. Finally, the performance of the proposed $v\omega$-AFC-FL control method is compared to the previous $v\omega$-AFC control scheme which has adopted the crude approximation technique in the estimation of **IN**.

3.2 Fuzzy Logic Applied to Robotics

Recently, some researchers have investigated the application of FL in the control of mobile robot. A number of fuzzy controllers are developed for steering control, as proposed by Hu and Yang (2003) as well as Antonelli *et al*. (2002). Some

of the others are used for the navigation control of mobile robot in the unknown working environment, with the presence of obstacles. For obstacles avoidance, the mobile robot should be equipped with sensors such as camera, sonar or infrared sensing devices. Sensors detect the existence of the obstacles and the location of the object in the workspace. In Aycard *et al*. (1997), actual orientation error or velocity is used as the inputs for the fuzzy membership function, while the outputs are velocity and heading angle. The difficulties of fuzzy design lie in the choice of appropriate inputs for fuzzification, the rules definition and defuzzification outputs (Aycard *et al*., 1997).

Mailah and Rahim (2000) have investigated a method for the estimation of inertia matrix for a robot arm, which is controlled by the active force control strategy (AFC) with embedded fuzzy logic mechanism. Fuzzy logic is incorporated into the control scheme to compute the inertia matrix (**IN**) of a robot arm intelligently, so that it can be utilized by the AFC mechanism in disturbance compensation. The inputs for the fuzzy membership function are the vector of joint angles (θ_1, θ_2), while the outputs are the elements of the estimated inertia matrix (IN_1, IN_2), for the first and second arm, respectively.

In this study, FL is applied to the system in order to obtain the 'optimal' values of the estimated inertia matrix similar to a study by Mailah and Rahim (2000). However, the inputs for the membership function in this mobile robot research are different from those described in their work. There are several potential variables that can be used as the input for fuzzy controller, namely linear and angular velocities, left and right wheel accelerations, trajectory tracking errors and the heading angle.

3.3 Fuzzy Logic Design

Mamdani method is widely accepted for expert knowledge capturing. It allows us to describe the expertise in more intuitive, more human-like manner. However, *Mamdani*-type fuzzy inference entails a substantial computational burden. On the other hand, *Sugeno* method is computationally effective and works well with

optimization and adaptive techniques, which makes it very attractive in control problems, particularly for dynamic nonlinear systems.

3.3.1 Definition of the Linguistic Variables

First of all, the inputs and outputs of the fuzzy controller system need to be identified. There are several potential variables as inputs the proposed fuzzy controller, namely acceleration, linear velocity, linear and angular velocities, trajectory tracking errors, heading angle and the estimated inertia matrix, **IN** as output. Figure 3.1 illustrates the configuration of the inputs and outputs of a fuzzy controller.

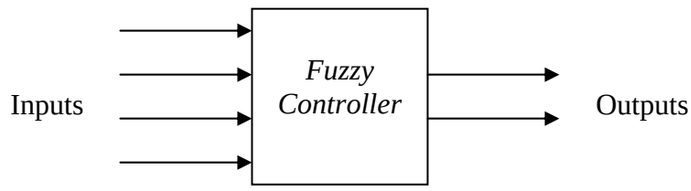


Figure 3.1: Inputs and outputs of the fuzzy controller

Figure 3.2 shows the comparison of trajectory tracking errors of the mobile robot considering different fuzzy inputs, namely acceleration and linear velocity, linear and angular velocities and trajectory tracking error and heading angle. These were tried on a control scheme to determine which combination produces the minimum error. It is observed that by using acceleration, linear velocity as the input variable for the fuzzy controller, the trajectory tracking error is the minimum. Therefore, in this study, the acceleration and linear velocity combination is chosen as the inputs for the fuzzy controller. The input linguistic variables are shown in Tables 3.1 and 3.2.

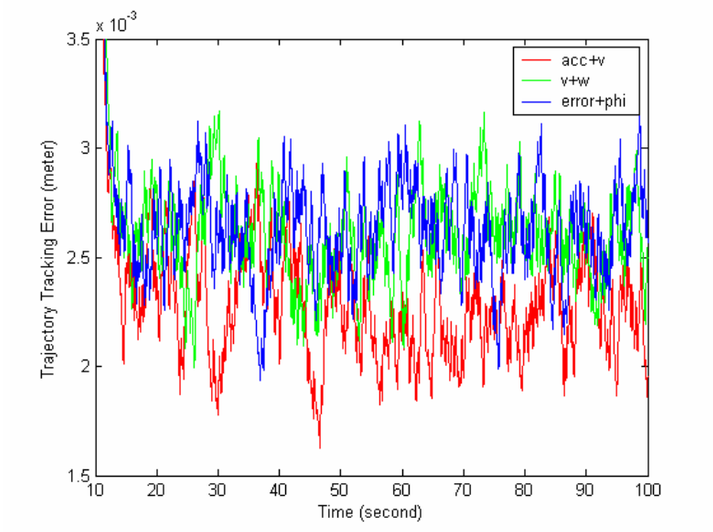


Figure 3.2: Comparison of trajectory tracking error using various inputs

Table 3.1: Acceleration

Linguistic Variable: *Acceleration*	
Linguistic Value	**Numerical Range**
Very Low (VL)	-2.0 to 0.0
Low (L)	-1.0 to 1.0
Medium (M)	0.0 to 2.0
High (H)	1.0 to 3.0
Very High (VH)	2.0 to 4.0

Table 3.2: Linear velocity

Linguistic Variable: *Linear Velocity*	
Linguistic Value	**Numerical Range**
Very Slow (VS)	-0.1 to 0.3
Slow (S)	0.1 to 0.5
Medium (M)	0.3 to 0.7
Fast (F)	0.5 to 0.9
Very Fast (VF)	0.7 to 1.1

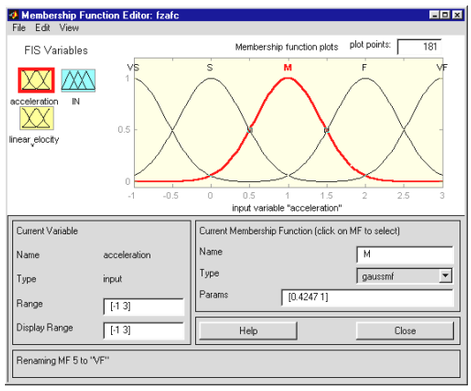
For **IN,** it is classified into following sets: *very small* (VS), *small* (S), *medium* (M), *big* (B) and *very big* (VB). The output linguistic variables are as shown in Table 3.3.

Table 3.3: Estimated inertia matrix, **IN**

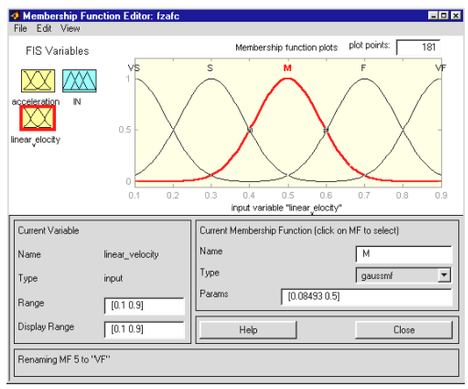
Linguistic Variable: **IN**	
Linguistic Value	**Numerical Range**
Very Small (VS)	-0.1 to 0.3
Small (S)	0.1 to 0.5
Medium (M)	0.3 to 0.7
Big (B)	0.5 to 0.9
Very Big (VB)	0.7 to 1.1

3.3.2 Determination of Fuzzy Set

There are various shapes of fuzzy set, but usually only triangle or trapezoid is chosen as the fuzzy sets, since it provides an adequate representation of the expert knowledge and at the same time it is simplifies significantly the computation prepossesses. The linguistic variables of inputs are presented in Figures 3.3(a) and (b), while the output membership function is shown in Figure 3.4.



(a) Acceleration



(b) Linear velocity

Figure 3.3: Inputs functions

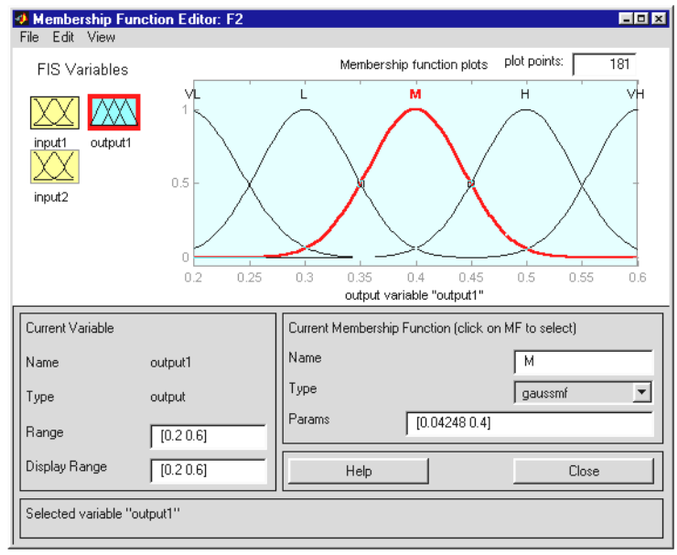


Figure 3.4: Estimated inertia matrix as the output function

3.3.3 Design of the Fuzzy Rules

To design the fuzzy rules, the knowledge or information about the behaviour of the relevant part of the proposed systems should be first obtained and later described in the form of fuzzy linguistic variables. The required knowledge for the design of the rules (solutions) can be gathered from other relevant sources such as books, computer databases, results from previous studies and observed human behaviour.

For FL method, the output range of the estimated inertia matrix is set from 0.1 to 0.9 kgm^2. Using a heuristic method, if the acceleration and the velocity are high or fast, the estimated inertia matrix of the dynamic system should be relatively high. On the contrary, if the acceleration and the velocity are low or slow, estimated inertia matrix of the dynamic system should be low. Therefore, the fuzzy rules can be created as shown in Figure 3.5.

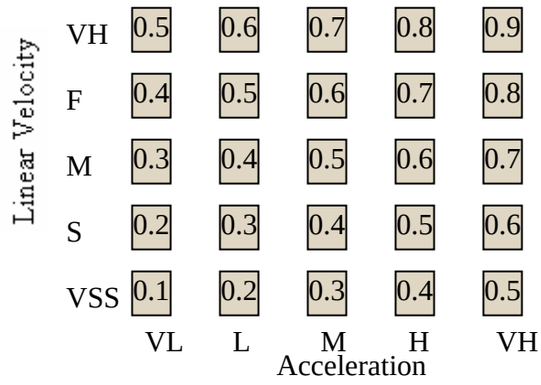


Figure 3.5: Fuzzy rules

The new output membership function is shown in Figure 3.6.

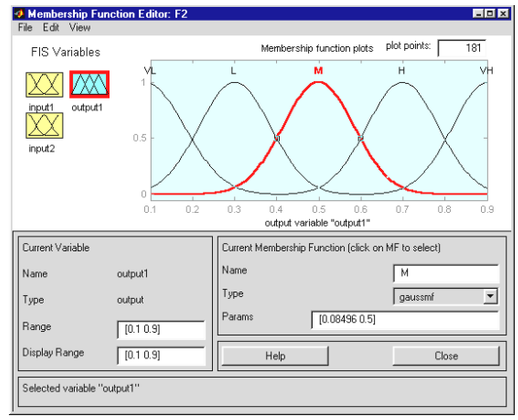


Figure 3.6: Extended estimated inertia matrix

In this research, there are two inputs and one output. In order to link the fuzzified inputs with the output, 5×5 fuzzy rules have been developed as follows:

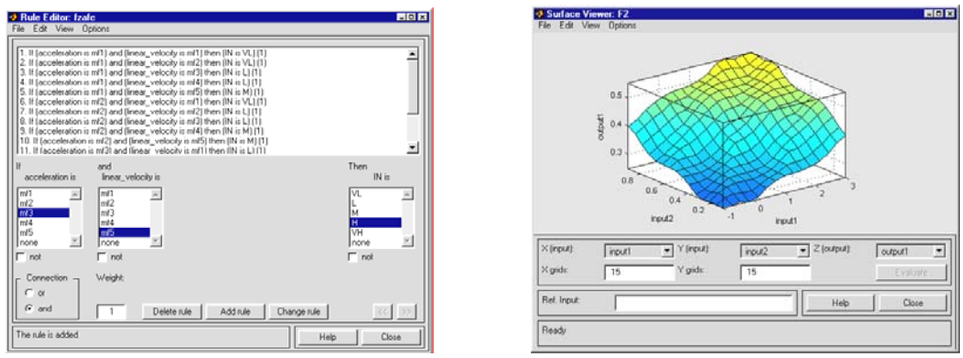
1. If (Linear Velocity is Very Slow) and (Acc. is Very Low) then (IN is Very Small)
2. If (Linear Velocity is Very Slow) and (Acc. is Low) then (IN is Very Small)
3. If (Linear Velocity is Very Slow) and (Acc. is Medium) then (IN is Small)
4. If (Linear Velocity is Very Slow) and (Acc. is High) then (IN is Small)

5. If (Linear Velocity is Very Slow) and (Acc. is Very High) then (IN is Medium)
6. If (Linear Velocity is Slow) and (Acc. is Very Low) then (IN is Very Small)

-
-
-

25. If (Linear Velocity is Fast) and (Acc. is Very High) then (IN is Very Big)

In this research, FL controller is developed using the Fuzzy Logic Toolbox® to be used with MATLAB and Simulink. The rule editor, surface viewer and rule viewer as shown in Figures 3.7 and 3.8 have been utilized for the determination of the characteristics of the proposed fuzzy controller.



(a) Rule editor (b) Surface viewer

Figure 3.7: Rule editor and surface viewer

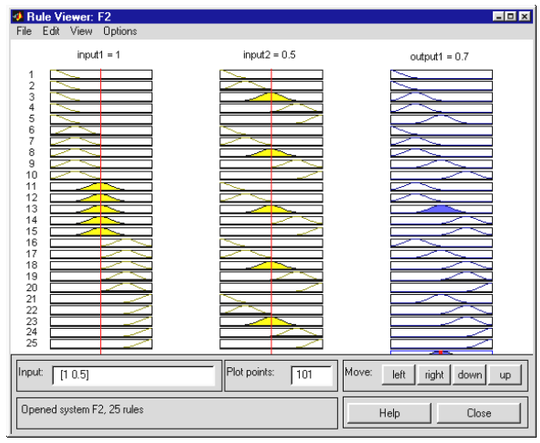


Figure 3.8: Rules viewer

3.4 Simulation

The simulation work is performed using the MATLAB/Simulink software packages. The block diagram for the $v\omega$-AFC-FL is shown in Figure 3.9. Compared with the previous control scheme, namely $v\omega$-AFC, $v\omega$-AFC-FL has been incorporated with FL algorithm which is mainly applied for the estimation of **IN**.

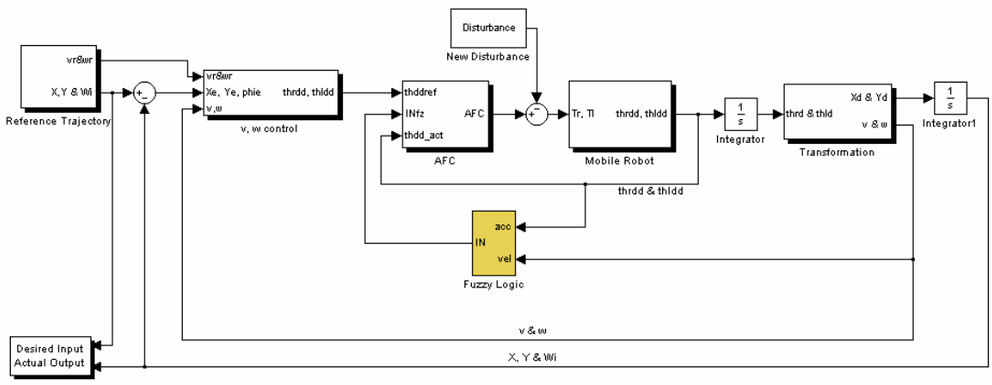
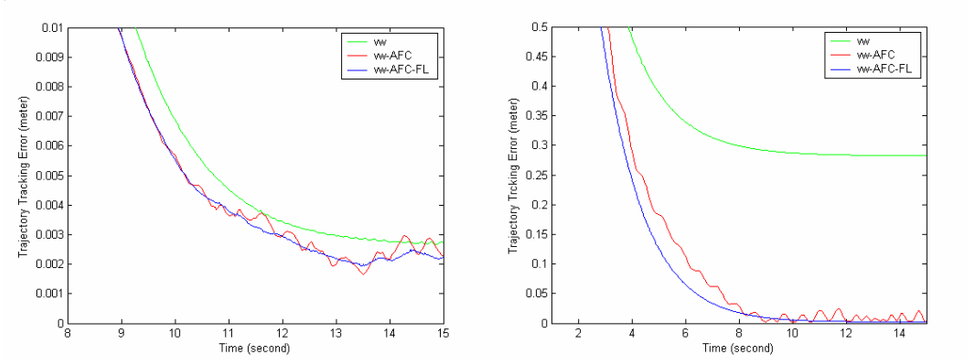


Figure 3.9: Simulink block diagram of $v\omega$-AFC-FL

3.5 Results and Discussion

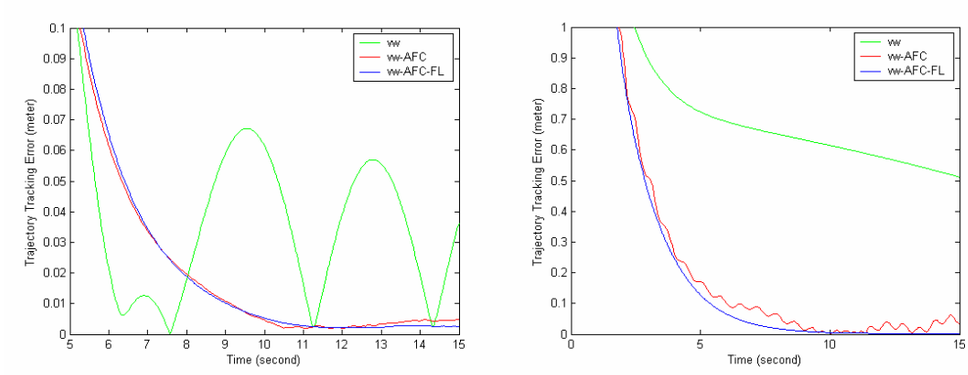
Simulation in this chapter is different with the previous simulation since it considers only quarter of desired circular trajectory. This is due to that the fact, the significant instability occurs only at the starting condition. Hereafter, the trajectory tracking error experiences minor changes only. Therefore the analysis will focus on the starting condition only.

Simulation studies on $v\omega$-AFC-FL controller are conducted in the existence of various disturbances, namely the trajectory tracking errors generated by $v\omega$-AFC controllers are then compared with $v\omega$ controller as shown in Figures 3.10 (a) to (d).



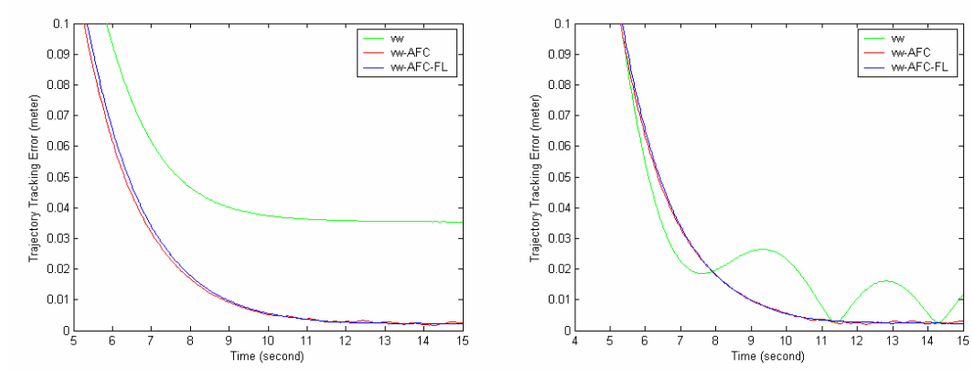
(a) F_o

(b) T_d



(c) F_s

(d) F_k



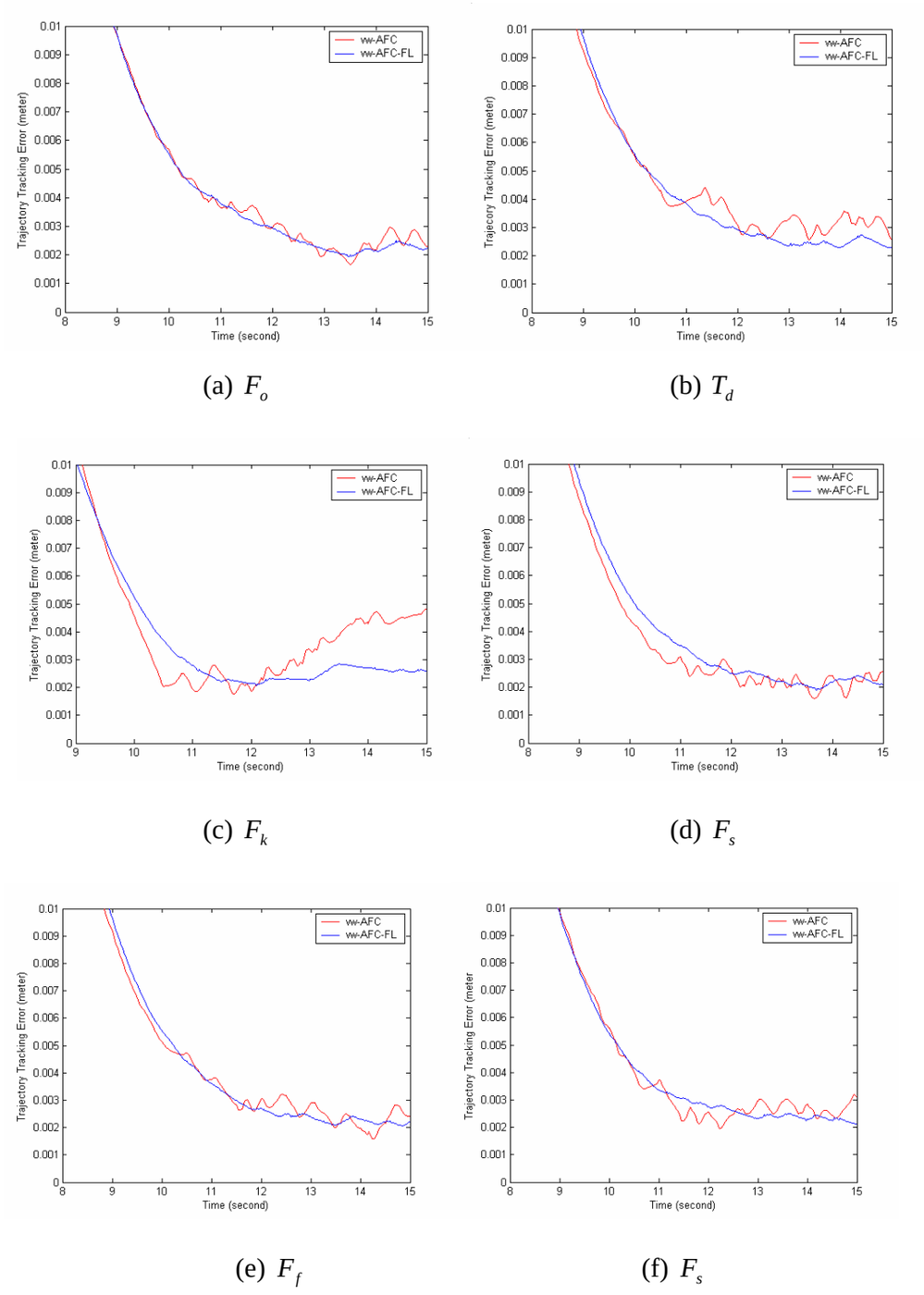
(e) F_f

(f) F_h

Figure 3.10: Comparison of the trajectory tracking errors in various disturbances

Figures 3.10 (a) to (f) show that the trajectory tracking error for $v\omega$ controller is generally very large, whilst those of $v\omega$-AFC and $v\omega$-AFC-FL schemes are much

smaller. Therefore, it is obvious that with the incorporation of AFC into the robotic system, the accuracy and stability of the system is greatly enhanced. Figures 3.11 (a) to (f) illustrate the comparison of results between $v\omega$-AFC and $v\omega$-AFC-FL in greater details.



(a) F_o (b) T_d

(c) F_k (d) F_s

(e) F_f (f) F_s

Figure 3.11: Comparison of the trajectory tracking errors between $v\omega$-AFC and $v\omega$-AFC-FL schemes

From Figure 3.11, it is observed that the trajectory tracking error for $v\omega$-AFC is generally very coarse and rugged, while the trajectory tracking error for $v\omega$-AFC-FL is relatively stable in smooth. It is clear that the $v\omega$-AFC-FL is more capable and smooth in the estimation of **IN** compared with $v\omega$-AFC.

3.6 Conclusion

The proposed $v\omega$-AFC-FL scheme has been developed, simulated and applied to the mobile robot. It performs excellently even under the influence of external disturbances. The FL mechanism computes **IN** automatically and continuously while the mobile robot is in operation. The main advantage of obtaining the estimated inertia matrix using FL is that it causes the fluctuation of the tracking error to considerably decrease compared to the crude approximation method showing that the AFC scheme is able to make the system robust and stable. The FL mechanism manages to compute **IN** automatically and continuously, while the mobile robot is commanded to track the referenced trajectory by driving the motor (and hence the robot) smoothly to the desired positions.

CHAPTER 4

VIRTUAL WHEELED MOBILE ROBOT SIMULATOR

4.1 Introduction

In this chapter, a virtual reality (VR) method is introduced and applied to the motion planning and control of a wheeled mobile robot (WMR). Research on the integration of VR with robotics is relatively new. Sensing the potentials of VR in robotics, a new era of research that is primarily concerned with the construction of the virtual simulator for robotic experimentation has evolved. In the simulator, the VR provides an advanced human-computer interface and this has greatly facilitated the study of robotics. In this chapter, the integration of VR into the simulation of the WMR will be discussed appropriately. The first section of the chapter touches on a description of the robot simulator and highlights the overall architecture of the proposed virtual WMR simulator. This is followed by the construction of the scene graph for the virtual environment (VE) of the mobile robot's workspace (layout). In addition, the elements within the scene graph will be also briefly discussed. The simulation study based on some of the theoretical concepts and principles derived from previous chapters are fully described in this chapter. Collision avoidance of obstacles is also discussed in the chapter based on a selected algorithm. The overall system is tested and evaluated considering the WMR to operate in a computer integrated manufacturing (CIM) environment.

4.2 Robot Simulator

For the experimentation of robotics, it is not necessary to have a physical robot and thus the hardware of the robot is actually incidental in this case (Schildt, 1987). In fact, it is the software that differentiates a robot from a machine. A robot simulator actually provides an imaginary world for the testing of the robot control algorithms. In general, a robot simulator should be instilled with the robotic-control algorithms, a database to store spatial information about the workspace, and a complete display of the motion-simulation results. In this research, the constructed virtual WMR simulator is equipped with the capabilities in path planning and motion control. Meanwhile, for the presentation of the simulation results, a specific VR technique has been applied in order to improve the analysis of the results. The overall architecture of the virtual WMR simulator is discussed in the following sections.

4.3 Overall Architecture of Virtual WMR Simulator

The construction of the virtual WMR simulator involves the aspects of motion path planning, motion control and the application of VR technique. Therefore, in order to ensure a systematic and effective simulation study to be carried out, the coordination and inter-relationship between all these subfields have to be carefully planned and implemented. The integration of all these fields of study for the construction of the virtual WMR simulator is as shown in Figure 4.1. A block diagram showing the interaction of the main components of the proposed simulator is illustrated in Figure 4.2.

At first, the map of the workspace for the virtual mobile robot is captured by a virtual camera that is installed within the virtual environment. The map (represented in bitmap form) is then interpreted and analyzed for the generation of the configuration space and the six elementary jumps graph. From this generated six elementary jumps graph, the path planner searches for a near-optimal time-independent collision-free path through A* heuristic search algorithm which is highlighted in **Appendix A**. Since the motion control algorithm requires the input of

the time-based reference trajectory, this geometrically planned collision-free path has to be transformed into a function of time. For this purpose, the concept of parametric cubic spline interpolation has been applied for the modelling of the trajectory planner.

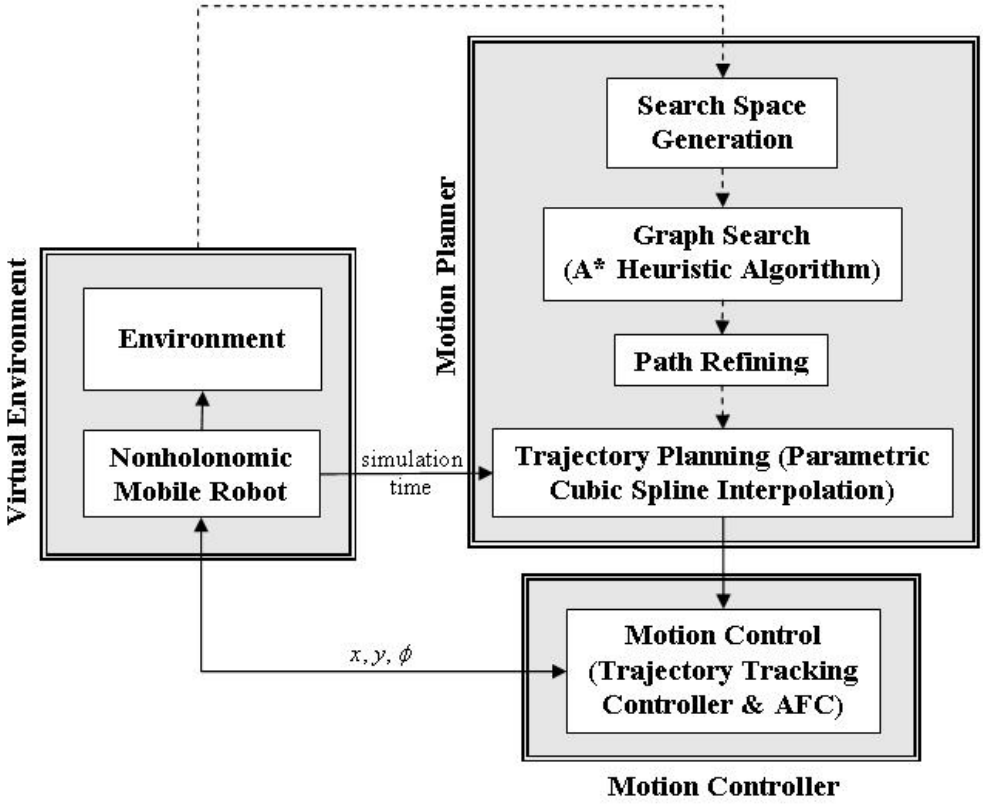


Figure 4.1: Interlinking of motion planning, motion control and VR

Considering the planned trajectory as the reference trajectory where the mobile robot is required to replicate (converge), the errors of the motion can be obtained through the subtraction of the reference states (q_{ref}, and $\dot{q}_{ref}$) from the actual states (q_{act}, and $\dot{q}_{act}$) of the mobile robot. The errors are then fed forward into the resolved acceleration control (RAC) component for the generation of the acceleration commands, $\ddot{q}_c$. Since the total DOF of the WMR is three, while controllable DOF is two, therefore the DOF for $\ddot{q}_c$ has to be stepped down to two.

The transformation of $\ddot{q}_c$ into the right and left wheel acceleration $\ddot{\theta}_{r,l}$ is as described in Chapter 2.

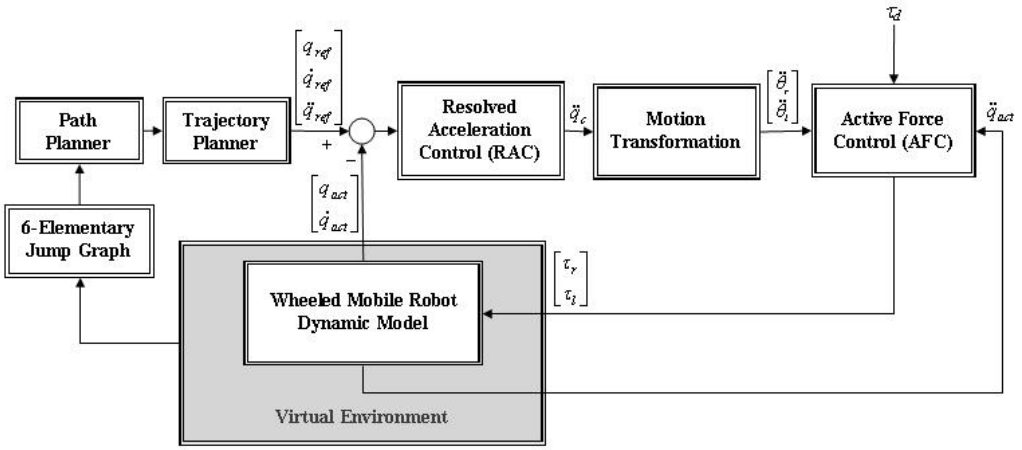


Figure 4.2: A block diagram of the virtual WMR simulator

From the calculated $\ddot{\theta}_{r,l}$ together with the disturbance torque, τ_d and actual acceleration, $\ddot{q}_{act}$, the active force control (AFC) will generate the control torques for the right and left motors of the mobile robot which will be fed into the WMR dynamic model as the actuating torques. The corresponding motions of the mobile robot are then integrated for the estimation of q_{act} and $\dot{q}_{act}$. Meanwhile, the geometry of the WMR within the virtual environment will also be transformed according to the current states, q_{act} of the mobile robot. Finally, q_{act} and $\dot{q}_{act}$ are fed back for control purposes.

4.4 Trajectory Planning

The path planner generates only a geometrical collision-free path and it is not ready for use by the motion controller which requires the time-based trajectory parameter as the input. Therefore, the geometrical collision-free path has to be

redefined into a time-based trajectory before it is fed into the motion controller for the trajectory tracking control. For this purpose, a trajectory planner has to be modelled in such a way that it is able to approximate the desired shape of the path curves into functions of time. Meanwhile the generated trajectory functions should also use less memory resource and able to offer easier manipulation of the collision-free path.

It is assumed that the geometrical collision-free path is actually a sequence of points in three dimensions (x, y, ϕ) through which the mobile robot is required to follow. In this research, the parametric cubic spline functions are applied for the description of the trajectory functions. This is because the parametric representation of curves in the form $x=x(t)$, $y=y(t)$, $\phi=\phi(t)$, is able to overcome the problems caused by the functional or implicit forms. Polynomials which are of lower-degree than cubic polynomials provide too little flexibility in controlling the shape of the path, while higher-degree polynomials can introduce unwanted 'wiggles' and also require more computational resources (Foley *et. al*, 1996). The general form of the parametric cubic spline function is given as

$$Q(t) = a_1 t^3 + a_2 t^2 + a_3 t + a_4 \tag{4.1}$$

With reference to equation (4.1), the parametric cubic polynomials that define a segment of the path, $Q(t) = [x(t)\ y(t)\ \phi(t)]$ are as follows:

$$\begin{aligned} x(t) &= a_{1x} t^3 + a_{2x} t^2 + a_{3x} t + a_{4x}, \\ y(t) &= a_{1y} t^3 + a_{2y} t^2 + a_{3y} t + a_{4y}, \\ \phi(t) &= a_{1\phi} t^3 + a_{2\phi} t^2 + a_{3\phi} t + a_{4\phi}, \ 0 \le t \le 1. \end{aligned} \tag{4.2}$$

By considering $q(t_k) = q_{tk}$, and $q\,'(t_k) = q\,'_{tk}$ and k is the number of segments of the curve, the coefficients, a_1 , a_2 , a_3 , and a_4 in Equation (4.1) can be expressed as (Cook and Ho, 1985):

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} \frac{2}{t_{k+1}^3} & -\frac{2}{t_{k+1}^3} & \frac{1}{t_{k+1}^2} & \frac{1}{t_{k+1}^2} \\ -\frac{3}{t_{k+1}^2} & \frac{3}{t_{k+1}^2} & -\frac{2}{t_{k+1}} & -\frac{1}{t_{k+1}} \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_k \\ q_{k+1} \\ q'_k \\ q'_{k+1} \end{bmatrix} \tag{4.3}$$

From Equation (4.3), q ’ has to be calculated first before the coefficients a_1, a_2, a_3 , and a_4 for each curve segment can be obtained. For the computer-controlled WMR, it is assumed that the initial and goal states of the mobile robot are zero. Therefore, through a geometrical path with n points, this standstill condition produces the boundary conditions,

$$q'_1 = q''_1 = q'_n = q''_n = 0 \tag{4.4}$$

Meanwhile, through the assumption of the continuity constraints between neighbouring spline segments as follows:

$$q''(t_{k+1}) = q''(t_k) \tag{4.5}$$

The calculation of q’ is given as:

$$[Mat]\,[q'] = [A_r] \tag{4.6}$$

where,

$$Mat = \begin{bmatrix} \frac{2}{t_3}+\frac{3}{t_2} & \frac{1}{t_3} & & & & \\ t_4 & 2(t_4+t_3) & t_3 & & & \\ & t_{k+2} & 2(t_{k+2}+t_{k+1}) & t_{k+1} & & \\ & & & \cdot & & \\ & & & \cdot & & \\ & & & & \frac{1}{t_{n-1}} & \frac{2}{t_{n-1}}+\frac{3}{t_n} \end{bmatrix}$$

$$q' = \begin{bmatrix} q'_2 \\ q'_3 \\ q'_{k+1} \\ \cdot \\ \cdot \\ q'_{n-1} \end{bmatrix}, \quad A_r = \begin{bmatrix} \frac{3}{t_3^2}(q_3 - q_2) + \frac{6}{t_2^2}(q_2 - q_1) \\ \frac{3}{t_3 t_4}\left[t_3^2(q_4 - q_3) + t_4^2(q_3 - q_2)\right] \\ \frac{3}{t_{k+1} t_{k+2}}\left[t_{k+1}^2(q_{k+2} - q_{k+1}) + t_{k+2}^2(q_{k+1} - q_k)\right] \\ \cdot \\ \cdot \\ \frac{3}{t_{n-1}^2}(q_{n-1} - q_{n-2}) + \frac{6}{t_n^2}(q_n - q_{n-1}) \end{bmatrix} \tag{4.7}$$

By taking the tri-diagonal characters of the coefficient matrix [*Mat*], the solution of Equation (4.6) can be obtained through *Gaussian* elimination to yield [*q* ’] which is then used for the calculation of the coefficients a_1, a_2, a_3, and a_4 in Equation (4.3). The velocity and acceleration of the mobile robot can be obtained through the respective first and second derivatives of Equation (4.1) as follows:

$$q\,'(t) = 3a_1 t^2 + 2a_2 t + a_3 \tag{4.8}$$

$$q\,''(t) = 6a_1 t + 2a_2 \tag{4.9}$$

4.5 Scene Graph of Virtual Environment

Virtual environment (VE) is actually composed of various objects, such as the lights, positional information, fogs and geometries. In order to manage all these objects systematically, the concept of scene graph is applied, where the whole structure of VE is maintained in a hierarchical structure (Engineering Animation, 1999) as shown in Figure 4.3. The actual scene graph program of the virtual environment for the CIM facility is shown in **Appendix B**.

This kind of structure resembles an upside-down tree since the root is at the top, and the branches and leaves are at the bottom. With reference to Figure 4.3, the *Root Node* is the ancestor for all other *Nodes*. The *Root Node* has several *Child Nodes* which are the direct descendant of it, such as the *Light Node* and *Group Node*.

Parent Node is defined as a *Node*'s direct ancestor, and every *Child Nodes* is allowed to have only one *Parent Node*. For example, at the second level of the hierarchical tree, the *Content Node*, and both of the *Separator Nodes* only have one single parent, that's the *Group Node*.

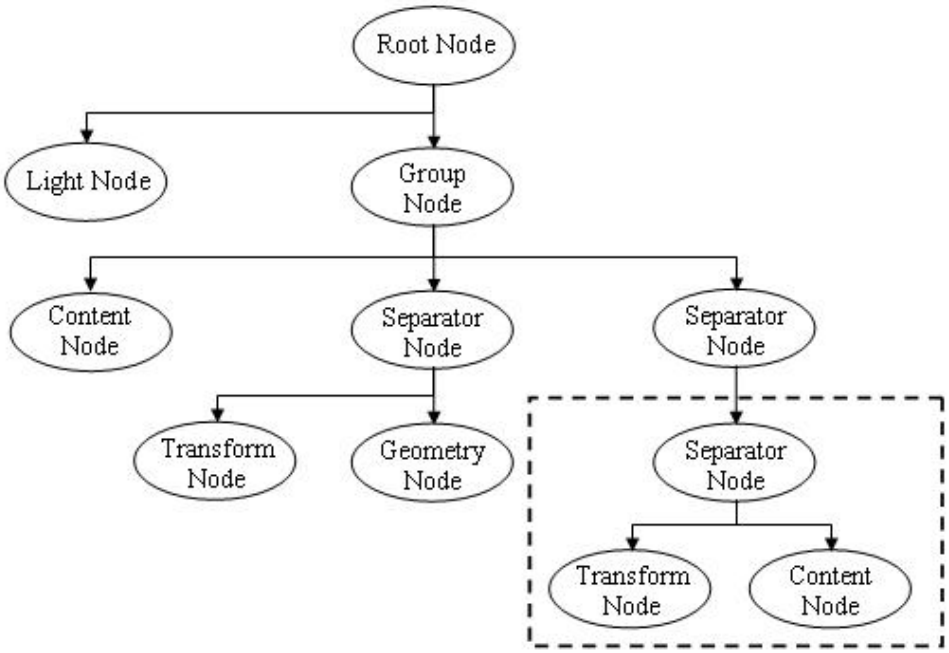


Figure 4.3: Hierarchical structure of the scene graph for a VE

Generally, all the objects in the virtual environment are represented as *Node*, which is the most fundamental element of the scene graph. The *Node* contains the information about the physical objects, and it can be categorized into *Content Node* and *Grouping Node*. *Content Node* holds the information about the VE and there are generally four types of *Content Nodes*: *Geometry Node*, *Light Node*, *Transform Node*, and *Fog Node*. *Geometry Node* contains the geometrical information about the physical and spatial states of an object. Meanwhile, the *Light Node* describes the lighting conditions of the virtual environment in terms of the light's position, intensity, ambient colour, and diffuse colour. On the other hand, the *Fog Node* holds the information about the characteristics of the fog within the VE. The fog is a special effect that simulates environmental conditions such as smoke, haze, and mist

which will obscure the view ability within the VE. The *Transform Node* contains the position or spatial transformation of the specified objects.

Unlike the *Content Node*, *Grouping Node* does not hold any direct information about the VE. However the presence of *Grouping Node* is essential in maintaining the hierarchical structure of the scene graph. Basically there are two types of *Grouping Node*: *Organizational Node* and *Procedural Node*. The main function of the *Organizational Node* is to group and encapsulate a set of *Nodes* in order to share the same common states together. The *Organizational Node* consists of *Group Node*, *Separator Node* and *Transform Separator Node*. *Procedural Node* acts in the same way as the *Organizational Node* but it contains additional information for the actuation of only one *child node* while deactivating the others. The *Procedural Node* consists of *Level-Of-Detail (LOD) Node* and *Switch Node*.

In the scene rendering, WorldToolKit (WTK) begins with the *Root Node* which is the entry point into the scene graph, and this is followed by all the *Nodes* attached to it from top to bottom and left to right order. When WTK encounters a *Node* with more than one *Child*, the first child's branch will be explored until all the *Child Nodes* within this specified *Node* is traversed completely. Figure 4.4 illustrates the procedures of VE rendering of the scene graph in Figure 4.3.

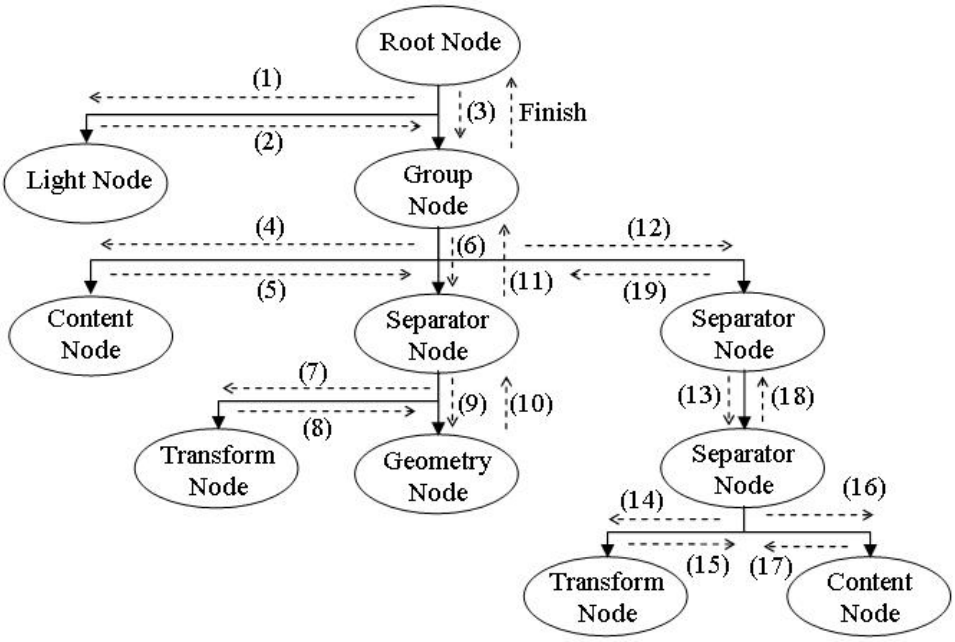


Figure 4.4: Procedures of VE rendering for a hierarchical structured scene graph

4.6 Simulation Setup

The virtual WMR simulator is constructed using Microsoft Visual C++ aided by the programming-ready mathematical functions that are provided by the MATLAB library. Besides, the incorporation of WTK has also reduced the programming burden in the construction of VE. For ease of program handling, the concept of object-oriented programming has been adopted in the study.

During the simulation, the WMR is required to plan or opt for a collision-free path which is then transformed into a time-based reference trajectory. The obtained reference trajectory is fed into the motion controller algorithm for the mobile robot trajectory tracking control. Meanwhile, the WMR is perturbed from its reference trajectory by the constant disturbance torques and the harmonic disturbance torques for the testing of the system's robustness and stability. The simulation results are finally used for the motion transformation of the virtual robot within the VE. A

number of assumptions, parameters and techniques used in the study have been made and chosen for the construction of the virtual WMR simulator. They are described accordingly as follows:

a) ***<u>Nonholonomic Motion Planning</u>***

Mobile robot perception sensors:	Assumed perfect models
Search space applied:	6-elementary jumps graph
Graph searching method:	A* heuristic search
Trajectory planning method:	Parametric cubic spline interpolation

b) ***<u>Initial and Final Conditions of WMR</u>***

Initial configuration of WMR: $q_i = \begin{bmatrix} x_i \\ y_i \\ \phi_i \\ \theta_{ri} \\ \theta_{li} \end{bmatrix} = \begin{bmatrix} 100.0 \\ 800.0 \\ 4.713 \\ 0.0 \\ 0.0 \end{bmatrix}$

Goal points:	(1350,700)
Goal orientation:	Unspecified

c) ***<u>WMR Parameters:</u>***

Driving wheel radius:	r = 0.15 m
Distance between wheels and axis of symmetry:	b = 0.75 m
Distance between P_o and P_c:	d = 0.03 m
Mass of vehicle:	m = 31.0 kg
Mass of platform:	m_c = 30.0 kg
Moment of inertia of vehicle:	I = 15.625 kgm^2
Moment of inertia of wheel:	I_w = 0.005 kgm^2

d) ***<u>RAC Parameters:</u>***

Proportional constant for e_x:	K_{px} = 200.0
Proportional constant for e_y:	K_{py} = 200.0
Proportional constant for e_ϕ:	$K_{p\phi}$ = 200.0

Derivative constant for $\dot{e}_x$: $K_{dx} = 30.0$

Derivative constant for $\dot{e}_y$: $K_{dy} = 30.0$

Derivative constant for $\dot{e}_\phi$: $K_{d\phi} = 30.0$

e) Active Force Control Parameters

Motor constant: $K_t = \begin{bmatrix} 0.263 \\ 0.263 \end{bmatrix}$ Nm/A

AFC constant: $K_c = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}$

Crude approximation of inertia matrix: $\mathbf{IN} = \begin{bmatrix} 0.2 \\ 0.2 \end{bmatrix}$ kgm^2

f) Applied Disturbances

Constant disturbance torque: $\tau_a = [3\ 3]$ Nm

Harmonic disturbance torque (right wheel): $\tau_{hr} = 3\sin(0.5t)+3$ Nm

Harmonic disturbance torque (left wheel): $\tau_{hl} = 3\sin(0.5t+\pi/2)+3$ Nm

g) Simulation Parameters for Collision-Free Trajectory Tracking

Tangential velocity: $Vel = 0.5$ m/s

Integration algorithm: ode45 (Dormand-Prince)

Simulation time steps: 0.1 s (not in real-time)

Before the actual execution of the simulation program, a heuristic method is applied for the estimation of the controller gains, K_{px}, K_{py}, $K_{p\phi}$, K_{dx}, K_{dy}, and $K_{d\phi}$. The obtained values of these gains are assumed to be suitable for the purpose of simulation study. The AFC constant, K_c is set to 1.0 for full AFC implementation. The constructed virtual WMR simulator requires tighter control of some of the more important parameters in order to ensure a more robust and precise operation of the robotic system within the VE. Initially, the virtual simulation is performed in an ideal

workspace without considering any disturbances. At a later stage, disturbances are modelled and applied to the system to test the robustness of the simulator.

4.7 Simulation Results and Discussion

Through the constructed virtual WMR simulator, the mobile robot has been experimented under three different kinds of loading conditions. At first the mobile robot is assumed to be working in an ideal workspace where it is free from all sorts of disturbances. Later, the mobile robot is exposed to the constant and harmonic disturbance torques individually. During simulation, the results for each simulation task are obtained, analysed and compared.

Figure 4.5 (a) illustrates the trajectory tracking of the virtual WMR which is free from disturbances. Figure 4.5 (b) and (c) show the trajectory tracking activities of the virtual WMR under the disturbances of τ_a and τ_h respectively. From these three graphs, it is obvious that the mobile robot is fully capable of planning for a global collision-free path which will then be used for the trajectory tracking purposes. With the incorporation of AFC scheme into the motion controller, the robustness of the robotic system has been greatly enhanced. The AFC scheme has successfully generated the compensation torques used for disturbance cancellation, provided that the estimated inertia matrix has been appropriately obtained.

With reference to Figures 4.5 (d) to (f), the track errors as well as the orientation errors for the virtual WMR are generally small and this small margin of errors has successfully kept the virtual WMR on track. However, it is obvious that the performance of the motion controller in this simulation degrades significantly. This is due to the fact that the reference trajectory adopted in this study is far more complex than the pre-calculated reference trajectories as stated in Equations (4.15) to (4.20). Besides, high value of K_p has an adverse effect on the stability of the robotic motion controller system resulting in rigorous fluctuation of the track errors. This might be resolved by increasing the value of K_d at the expense of degrading the precision of the trajectory tracking control.

From Figures 4.5 (d) to (f), the most critical period is observed to occur within the time range 15 – 20 s. At this stage, the mobile robot experiences the largest change of heading direction. As a result, this causes the track errors for x to increase drastically. However, due to nonholonomic constraints, the track errors for y experiences a slight change only, considering the virtual WMR satisfies the pure rolling and non-slipping conditions.

Figures 4.5 (g) and (h) illustrate the computed torques for the right and left motors respectively. With the incorporation of AFC into the motion controller scheme, the system is able to compute the necessary applied control torque that is required to cancel out the disturbances. However, the computed torques increase drastically whenever the virtual WMR is required to change its heading direction as observed in Figures 4.5 (g) and (h) at a steady state condition. This consequently leads to the saturation of the computed torques, and thus results in prolonged convergent time of the motion controller.

The sharp increase of the computed torque is due to the abrupt change of inertia of the WMR plus the effect of friction whenever the robot is required to turn suddenly. One way to counter this problem is to ensure a more gradual turning of the WMR through proper path planning and control. This also involves appropriate acquisition of WMR and controller parameters particularly the good estimation of the **IN** (inertia matrix) parameter in the AFC loop and the controller gains.

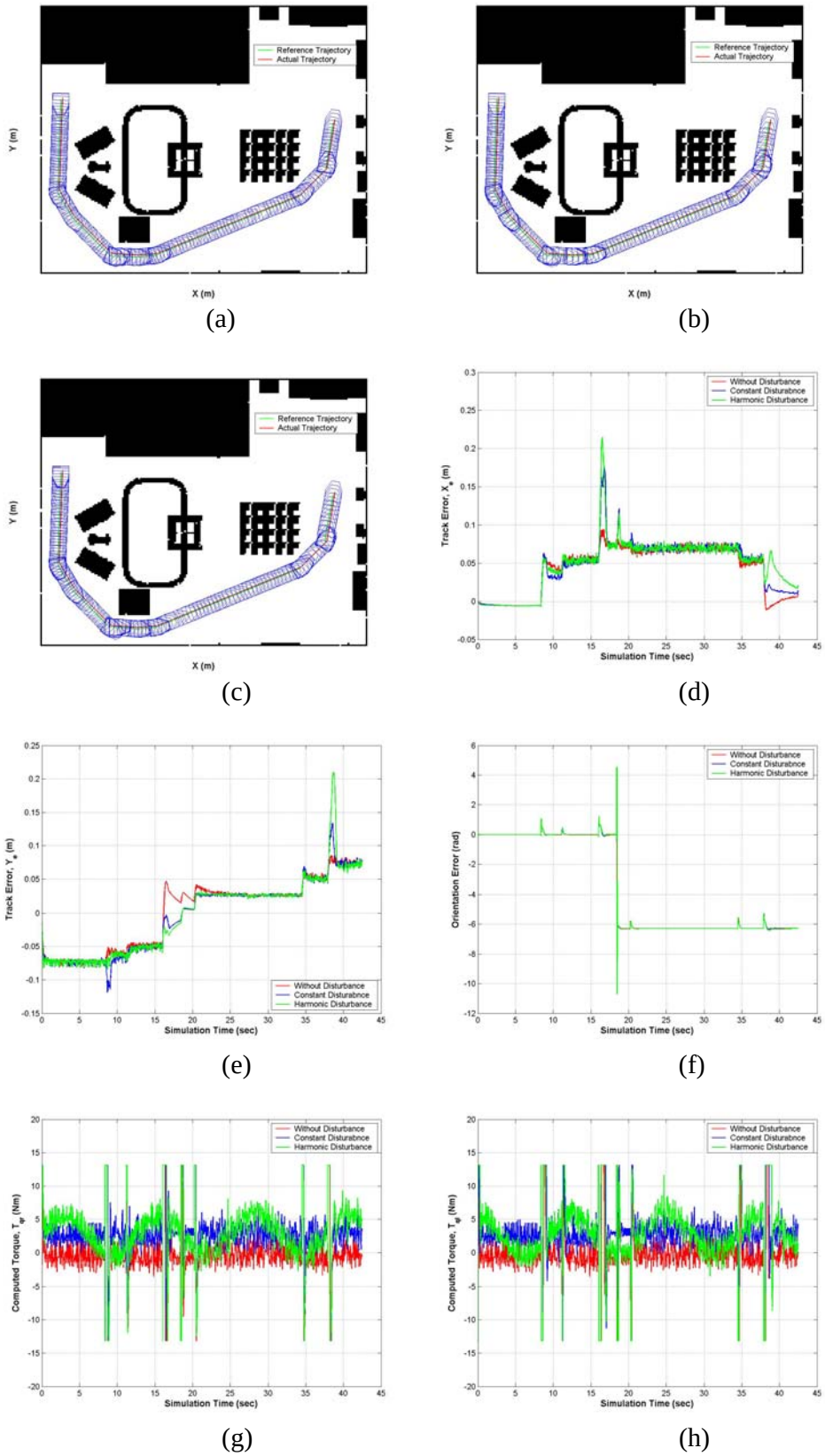
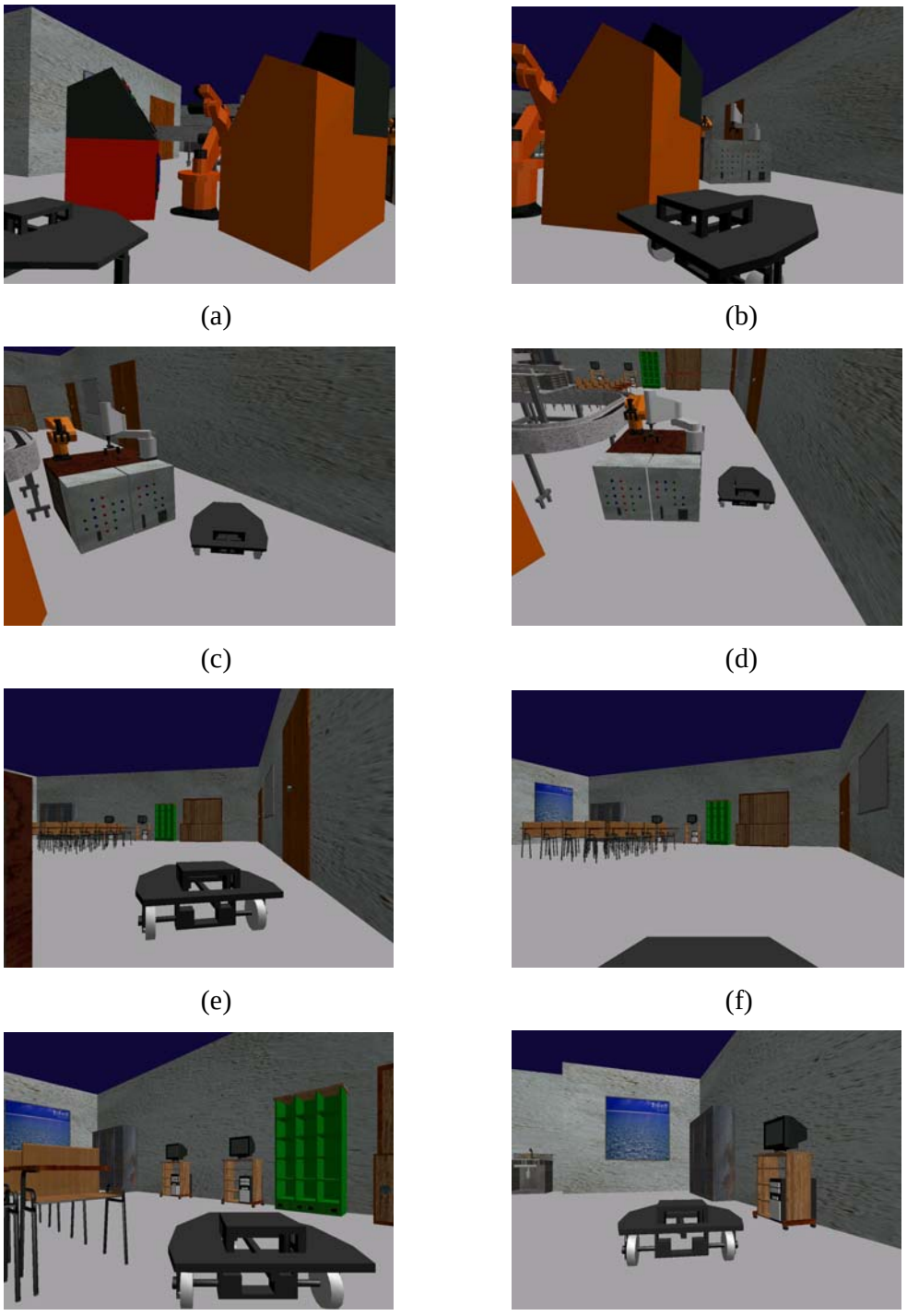


Figure 4.5: Performance of the simulator under various loading conditions

To promote better understandings of the simulation results involving the physical workspace of the mobile robot, the technology of VR has been applied. Figures 4.6 (a) – (h) illustrate several views of the virtual WMR simulator.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 4.6: Several views of the constructed virtual WMR simulator

By displaying the simulation results graphically, the interactivity between the virtual WMR and its VE can be analyzed in greater depth. Therefore, the knowledge gained through the simulation can then be used for the experimentation of physical WMR in real world.

4.8 Conclusion

The developed virtual WMR simulator has been proven to be very effective as the results of the experiments suggest. It is observed that the mobile robot is able to autonomously plan for a global collision-free path which is then transformed into a time-indexed trajectory through the proposed trajectory planner algorithm. With this generated trajectory, the proposed motion controller has successfully driven the virtual WMR to follow the reference path although it is disturbed by a number of introduced ‘noises’. The AFC scheme is found to be robust and effective in the generation of the compensated torques required for the control action. On the whole, it can be deduced that the main contribution of the developed virtual WMR simulator is that it has successfully provided a useful platform (through the construction of an imaginary world) for the testing of robot control algorithms. This has direct consequence on the possible reduction of the experimentation costs as well as the fact that it may serve as a training environment (e.g. on safety aspect and layout planning) for the prospective researchers/users.

CHAPTER 5

RESOLVED ACCELERATION CONTROL AND ACTIVE FORCE CONTROL (RACAFC) OF MOBILE MANIPULATOR

5.1 Introduction

This chapter discusses the proposed *Resolved Acceleration Control and Active Force Control* (RACAFC) scheme for motion control of mobile manipulator (MM) whose kinematic and dynamic models have been established in Chapter 2. The robot is a differentially driven wheeled mobile platform with a two-link planar arm mounted on top of the platform. A simplified coordinate (x, y) and heading angle (φ) of a nonholonomic mobile manipulator motion control using a class of resolved acceleration control combined with an active force control strategy is discussed. The RAC part is used to manipulate the kinematic components while the AFC scheme is implemented to compensate the dynamic effects including the bounded known/unknown disturbances and uncertainties. For convenience, the proposed scheme is called MM-RACAFC (*Mobile Manipulator – Resolved Acceleration Control and Active Force Control*), while the scheme without AFC is known as MM-RAC (*Mobile Manipulator – Resolved Acceleration Control*).

.

5.2 Proposed MM-RACAFC Scheme

The proposed scheme is divided into two categories, i.e., RAC that deals with the kinematic, and AFC that deals with the dynamics. The RAC implements position, velocity and acceleration signals with the referenced signals pertaining to the robot moving in a *Cartesian* coordinate also defined accordingly. Figure 5.1 shows the outline of the proposed scheme. From the figure, the proposed RAC is designed to provide the stability of the system performance and the convergence of the system's error. The AFC is proposed to specially compensate the non-linearity of the dynamics including disturbances. Even in fact the RAC is a converge (negative) feedback control that able to stabilize the system and converging the general error occurred but however the high non-linearity would be relatively uncompensated due to the performances limitation of the position & speed control (only) of the RAC (Muir and Neumann, 1990, Musa, 1998, Campa, et. al., 2001). Therefore, the proposed AFC is specially designed to solve this problem.

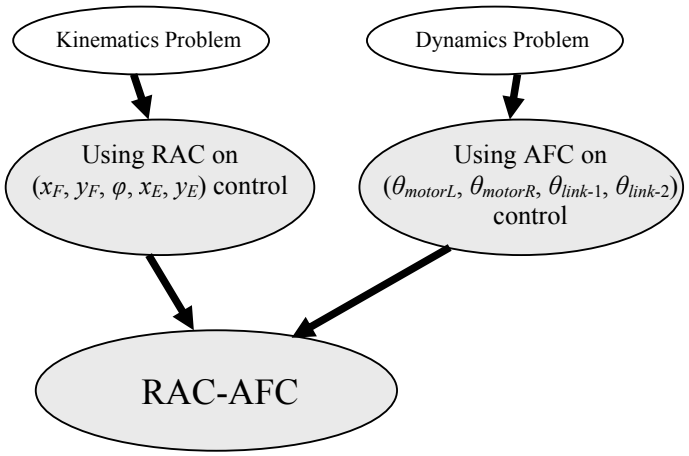


Figure 5.1: Outline of the proposed MM-RACAFC scheme

The proposed RACAFC controller is made up of two controllers that could be theoretically designed independently. The complete proposed MM-RACAFC scheme is shown in Figure 5.2.

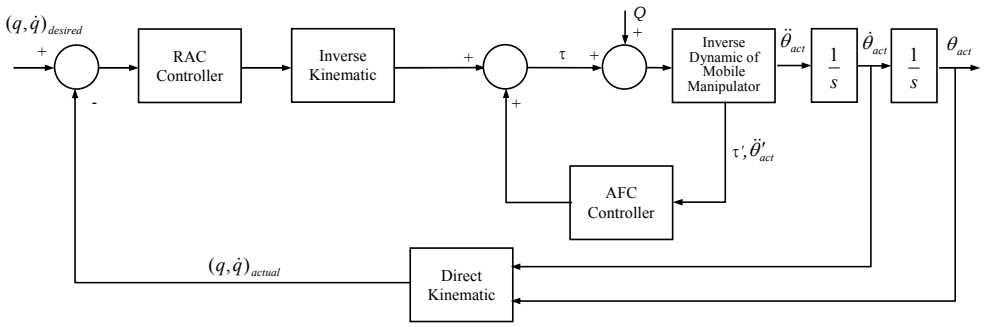


Figure 5.2: The proposed MM-RACAFC scheme

From Figure 5.2, the AFC is serially configured to the RAC so that basically the RAC can be tested without the AFC. In this case of without AFC the output of RAC (after transformed to angles domain of actuators) can be directly connected to the actuators (motors). For the RAC only the scheme can be redrawn as shown in Figure 5.3.

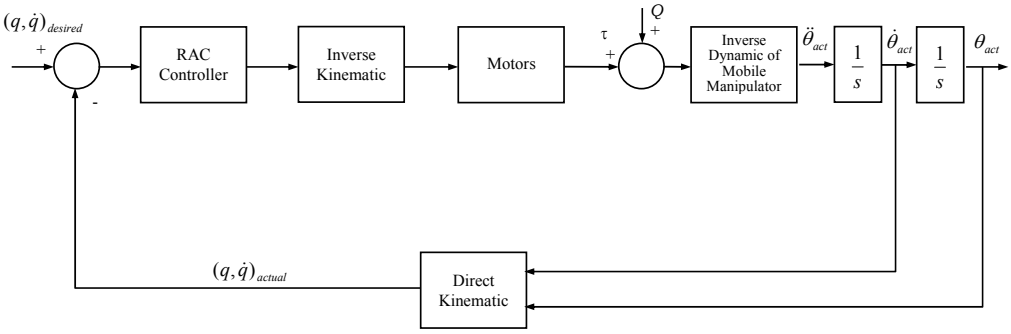


Figure 5.3: The MM-RAC scheme

For comparative study the scheme as shown in Figure 5.3 was also simulated and the results would be compared to the results on the scheme as depicted in Figure 5.2. Note that the two schemes would be represented in one schematic simulation diagram (will be discussed later).

The RAC part can be described as shown in Figure 5.4.

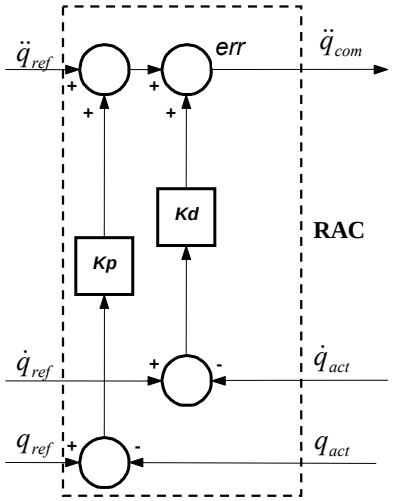


Figure 5.4: The RAC controller

From Figure 5.4, for q generalized coordinates of $q = [x_F, y_F, \varphi, x_E, y_E]^T$ in real number $\Re^n$ the output equation of the RAC can be written as:

$$\ddot{q}_{com} = \ddot{q}_{ref} + Kd(\dot{q}_{ref} - \dot{q}_{act}) + Kp(q_{ref} - q_{act}) \qquad (5.1)$$

The AFC controller in Figure 5.2 operates in acceleration mode. Its input is the transformed RAC output in angles domain of the motors. Figure 5.5 shows the schematic diagram of the AFC.

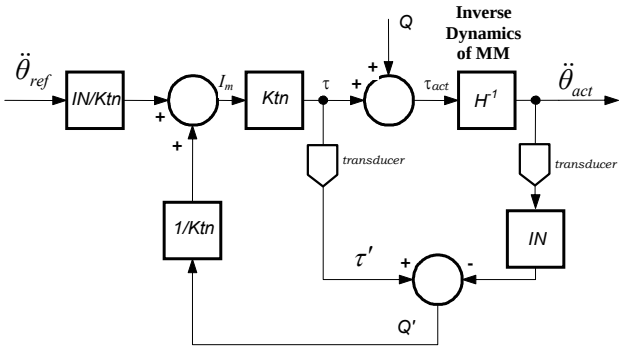


Figure 5.5: The proposed AFC controller

In Figure 5.5, **IN** is the mass moment of inertia matrix estimator (multiplier constant or function), *Ktn* is the motor constant, I_m is the motor current, τ is the actuated motor's torque, τ_{act} is the actual total applied motor's torque, τ' is the measured actuated motor's torque, Q is the bounded known/unknown disturbances, and Q' is the estimated disturbances. τ' can be simply measured using a torque sensor or by measurement of the actual motor current multiplied with the motor constant.

With reference to Figure 5.5, the simplified dynamic model of the system can be expressed as:

$$\tau_{act} = \tau + Q = I(\theta)\ddot{\theta}_{act} \tag{5.2}$$

where $I(\theta)$ is the mass moment of inertia of the wheels and arms, and θ is the angle at each wheel or joint, $\ddot{\theta}_{act}$ is the angular acceleration of the moving body.

Also from Figure 5.5, a measurement of Q' (i.e., an estimate of the disturbances, Q) can be obtained such that:

$$Q' = \tau' - \ddot{\theta}'_{act} IN' \tag{5.3}$$

where the superscript $'$ denotes a measured or estimated quantity. The torque τ' can be measured directly using a torque sensor or indirectly by means of a current sensor. Thus, Equation (5.3) can also be written as:

$$Q' = I'_m Ktn - \ddot{\theta}'_{act} IN' \tag{5.4}$$

where $I_m^{'}$ is the measured motor current.

For the AFC system in Figure 5.5, the actuated motor's torque can be written as:

$$\tau = \left(\frac{\ddot{\theta}_{ref} IN'}{Ktn} + \frac{Q'}{Ktn} \right) Ktn \tag{5.5}$$

Substituting Equation (5.4) into Equations (5.5) and (5.2), we obtain

$$\begin{aligned} \tau &= \left(\frac{\ddot{\theta}_{ref} IN'}{Ktn} + \frac{\left(I'_m Ktn - \ddot{\theta}'_{act} IN' \right)}{Ktn} \right) Ktn \\ &= \ddot{\theta}_{ref} IN' - \ddot{\theta}'_{act} IN' + I'_m Ktn \\ &= IN' \left(\ddot{\theta}_{ref} - \ddot{\theta}'_{act} \right) + I'_m Ktn \end{aligned} \tag{5.6}$$

and

$$\tau_{act} = IN' \left(\ddot{\theta}_{ref} - \ddot{\theta}_{act} \right) + I'_m Ktn + Q \tag{5.7}$$

Equation (5.6) is known as the AFC controller's output equation. It has been ascertained that if the measured or estimated values of the parameters in the equation were appropriately acquired, a very robust system that totally rejects the disturbances is achieved (Hewit and Burdess, 1981).

Finally, the proposed MM-RACAFC can be redrawn in one schematic diagram as shown in Figure 5.6.

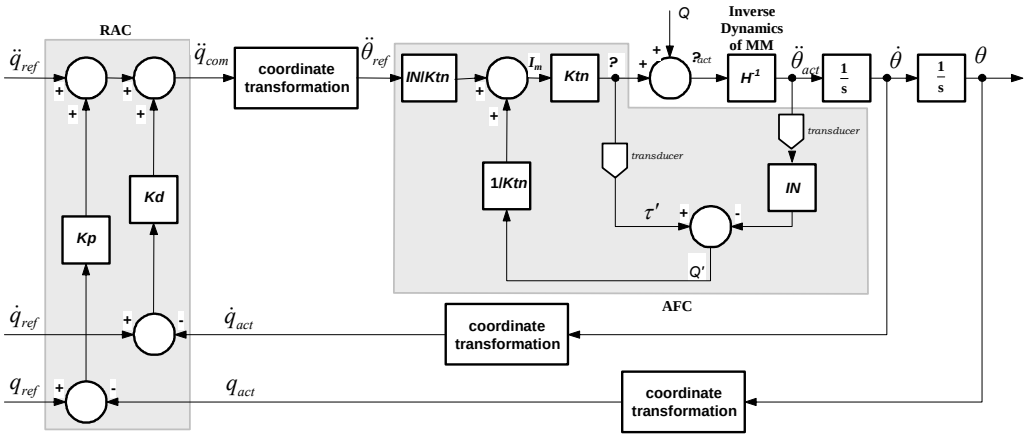


Figure 5.6: Schematic diagram of the proposed MM-RACAFC

In Figure 5.6, the subscripts of *ref*, *com* and *act* denote the *reference*, *command* and *actual* respectively. Considering $q = [x_F, y_F, \varphi, x_E, y_E]^T$ generalized coordinates, output equation of the RAC, $\ddot{q}_{com} = [\ddot{x}_{Fcom}, \ddot{y}_{Fcom}, \ddot{\varphi}_{Fcom}, \ddot{x}_{Ecom}, \ddot{y}_{Ecom}]^T$ consists of five controllers' output equations, i.e.:

$$\ddot{x}_{Fcom} = \ddot{x}_{Fref} + Kd(\dot{x}_{Fref} - \dot{x}_{Fact}) + Kp(x_{Fref} - x_{Fact}) \tag{5.8}$$

$$\ddot{y}_{Fcom} = \ddot{y}_{Fref} + Kd(\dot{y}_{Fref} - \dot{y}_{Fact}) + Kp(y_{Fref} - y_{Fact}) \tag{5.9}$$

$$\ddot{\varphi}_{Fcom} = \ddot{\varphi}_{ref} + Kd(\dot{\varphi}_{ref} - \dot{\varphi}_{act}) + Kp(\varphi_{ref} - \varphi_{act}) \tag{5.10}$$

$$\ddot{x}_{Ecom} = \ddot{x}_{Eref} + Kd(\dot{x}_{Eref} - \dot{x}_{Eact}) + Kp(x_{Eref} - x_{Eact}) \tag{5.11}$$

$$\ddot{y}_{Ecom} = \ddot{y}_{Eref} + Kd(\dot{y}_{Eref} - \dot{y}_{Eact}) + Kp(y_{Eref} - y_{Eact}) \tag{5.12}$$

where the subscript terms *ref*, *act* and *com* are the input reference, actual output and command respectively. *Kp* and *Kd* are the proportional and derivative gains respectively.

The AFC part is composed of four output equations in which each equation is related to each of the motor, i.e.:

$$\tau_1 = IN_1'\left(\ddot{\theta}_{1ref} - \ddot{\theta}_{1act}'\right) + I_{m1}' Ktn_1 \tag{5.13}$$

$$\tau_2 = IN_2'\left(\ddot{\theta}_{2ref} - \ddot{\theta}_{2act}'\right) + I_{m2}' Ktn_2 \tag{5.14}$$

$$\tau_L = IN_L'\left(\ddot{\theta}_{Lref} - \ddot{\theta}_{Lact}'\right) + I_{mL}' Ktn_L \tag{5.15}$$

$$\tau_R = IN_R'\left(\ddot{\theta}_{Rref} - \ddot{\theta}_{Ract}'\right) + I_{mR}' Ktn_R \tag{5.16}$$

where the subscripts 1, 2, *L*, and *R* denote motors of the link-1, link-2, left and right wheels respectively.

5.3 Proposed RACAFC for Mobile Platform Section

The successful implementation of AFC to the mobile manipulator partly depends on the feasibility of AFC applied to the mobile platform which has nonholonomic constraints. Before discussing the simulation on the proposed MM-RACAFC, an investigation into the RACAFC scheme applied to the mobile platform is firstly presented in the following paragraphs.

Differentially Driven Mobile Robot (DDMR) normally employs two independently driven wheels with a common axis and a castor to add to the stability of the platform. This mobile platform can be classified as a nonholonomic system such that the nonholonomic constraint cannot be expressed as time derivatives of some functions of the generalized coordinates. The use of DDMR as a mobile robotic system has been considered a challenging research area that is rigorously being studied by many researchers, either in kinematics or dynamics domain. In the proposed MM-RACAFC scheme this type of DDMR is used as platform of the mobile manipulator.

A number of nonholonomic mobile robot tracking control methods has been proposed: velocity and heading angle (v, ω) control (Kanayama, *e. al*., 1990), intelligent steering control and path planning using genetic algorithm (Yasuda and Takai, 2001) and an improved (v, ω) control with exponential stabilization using *Lyapunov*'s stability technique (Pourboghrat, 2002). The models used in the cited works are more concerned about the kinematics and the studies mostly relate to establishing the mobile robot's stability under varied initial conditions and trajectory input functions. Other researchers focus on the robust motion control such as using backstepping kinematics to dynamics method (Fierro and Lewis, 1995), (v, ω) control converted to torque control combined with neural network method (Fierro and Lewis, 1998) and adaptive control (Fukao, *et al*., 2000). Their works mainly centre around the techniques on how to manage the robot dynamics and cancel or compensate the disturbances. However, they did not address the computational factor when the methods were to be implemented in real robot programming implying that the methods are quite complex to be directly written in a compact program.

As an initial study, a simplified coordinates (x, y) and heading angle (φ) of nonholonomic mobile robot motion control using resolved acceleration control (RAC) combined with active force control (AFC) is discussed. The RAC that was initiated by Luh*, et al.* (1980) is a simple but yet powerful acceleration mode control method that could improve the existing conventional servo control scheme. By employing the RAC-based x and y control, instead of the v control, the proposed control scheme would have a more flexible position and speed control. This flexibility is gained by employing the input references of position, velocity and acceleration simultaneously. To overcome the dynamics problem related to effect of disturbances and uncertainties, an active force control scheme, firstly introduced by Hewit and Burdess (1981), was incorporated into the proposed RAC scheme. The complete control scheme applied to the DDMR is called as DDMR-RACAFC.

The proposed DDMR-RACAFC consists of two controllers that could be theoretically designed independently. Firstly, the RAC part consists of three controllers output equations defined as:

$$\ddot{x}_{Fe} = \ddot{x}_{Fref} + Kd(\dot{x}_{Fref} - \dot{x}_{Fact}) + Kp(x_{Fref} - x_{Fact}) \tag{5.17}$$

$$\ddot{y}_{Fe} = \ddot{y}_{Fref} + Kd(\dot{y}_{Fref} - \dot{y}_{Fact}) + Kp(y_{Fref} - y_{Fact}) \tag{5.18}$$

$$\ddot{\varphi}_{e} = \ddot{\varphi}_{ref} + Kd(\dot{\varphi}_{ref} - \dot{\varphi}_{act}) + Kp(\varphi_{ref} - \varphi_{act}) \tag{5.19}$$

where for $q = [x_F, y_F, \varphi]^T$ and the subscript terms *ref*, *act* and *e* are the input reference, actual output and error respectively. *Kp* and *Kd* are the proportional and derivative gains respectively. For application of the RAC scheme only, the controller output parameters with subscript *e* could be directly connected to the actuators input. The AFC part was designed to operate in the acceleration mode of each motor at the angular side of each wheel. The proposed AFC scheme is shown in Figure 5.7.

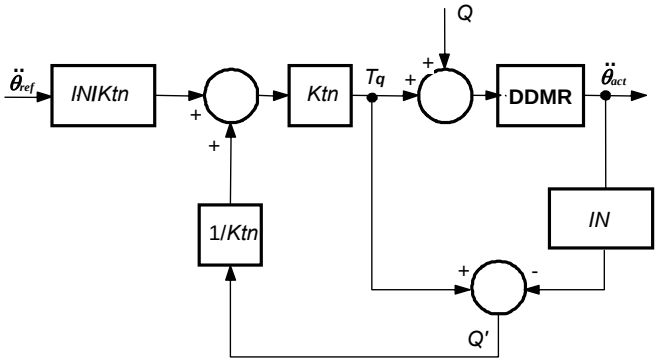


Figure 5.7: The proposed AFC applied to the DDMR

Q indicates the bounded (known or unknown) disturbances, *Ktn* is the motor constants (left and right), **IN** is the estimated inertia matrix, and *Q'* is the estimated disturbances calculated by the AFC algorithm. **IN** should be adjusted properly to trigger the disturbances cancellation process. The simulation of DDMR-RACAFC was then performed. The advantage of using the RACAFC scheme to perform the motion control with capabilities of compensating the effects of disturbances shall be rigorously investigated and presented in subsequent sections.

5.4 Simulation

The simulation consists of three schemes, i.e., RACAFC for mobile platform (DDMR-RACAFC), MM-RAC, and MM-RACAFC. The first scheme was used to test the convergences and controllability of the RAC combined to AFC in the case of nonholonomic constraint in the mobile platform. The results would be then applied to the proposed MM-RACAFC. The second simulation, i.e., MM-RAC was performed to test the performance of the RAC scheme without AFC in the case of mobile manipulator. The third, as the main proposed scheme in this chapter was simulated and then compared to the results of the MM-RAC scheme. All the schemes are simulated using MATLAB and Simulink software package.

5.4.1 Simulation Parameters

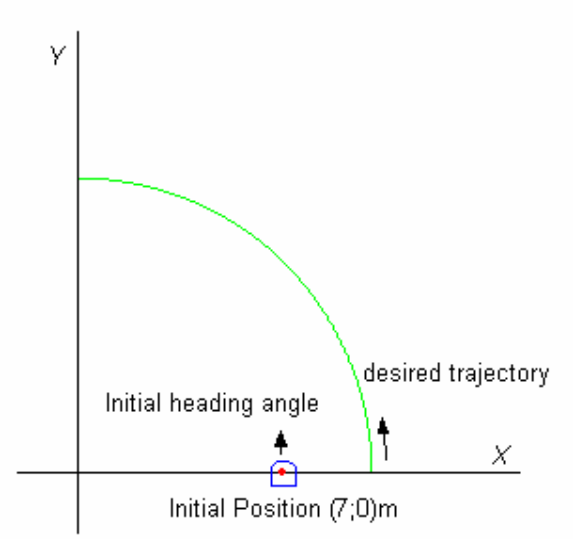
The simulations of MM-RAC and MM-RACAFC use same parameters for the robot's physical properties, task, the defined disturbances, the main control parameters (i.e., *Kp*, and *Kd*), and the method of simulation and its parameters. Tables 5.1, 5.2 and 5.3 show the specifications of the mobile manipulator, the prescribed trajectory (task) and the disturbances respectively. Data of the control parameters will be described later related to the simulation results.

Table 5.1: Specifications of mobile manipulator

Parameters	Mobile Platform	Manipulator
Type	Differentially-Driven Mobile Robot (left & right)	Two-link Planar Arm
Mass of platform	40 kg	
Mass of wheel	1 kg (each)	
Half width of platform	0.4 m	
Mass moment of inertia of platform about vertical axis	11.525 kg.m^2	
Mass moment of inertia of wheel about wheel diameter	0.05 kg.m^2	
Mass moment of inertia of wheel about wheel axis	0.05 kg.m^2	
Radius of wheel	0.11 m	
Distance of point *G* to *F*	0.3 m	
Mass of link-1 (plus motor at joint 2)		3.25 kg
Length of link-1		0.4 m
Length of link-1 from joint-1 to centre of gravity		0.24 m
Mass moment of inertia of link-1 about joint-1 axis		0.0433 kg.m^2
Mass of link-2 (plus gripper)		1.25 kg
Length of link-2		0.38 m
Length of link-2 from joint-2 to centre of gravity		0.21 m
Mass moment of inertia of link-2 about joint-2 axis		0.0151 kg.m^2
Motor constant of joint-1		0.0625 Nm/Ampere
Motor constant of joint-2		0.0371 Nm/Ampere
Motor constant of left wheel	0.0625 Nm/Ampere	
Motor constant of right wheel	0.0625 Nm/Ampere	
Gear ratio of Motor-1		100:1
Gear ratio of Motor-2		200:1
Gear ratio of Motor-L	100:1	
Gear ratio of Motor-R	100:1	

Table 5.2: The prescribed trajectories

Parameters	Mobile Platform /Vehicle only	Mobile Manipulator
Type of track	**See Figure 5.8.**	**See Figure 5.9.**
Radius of vehicle moving	10 m	10 m
Velocity of vehicle	0.4 m/s	0.2 m/s
Initial orientation (heading angle)	$\pi/2$ radian	$\pi/2.4$ radian
Radius of "TIP" function		0.14 m
Angular speed of arm (TIP)		0.06 m/s
Vehicle's initial position (x, y)	(7;0)m	(10;0)m
Tip's initial position (x, y)		(10.55;0.35)m



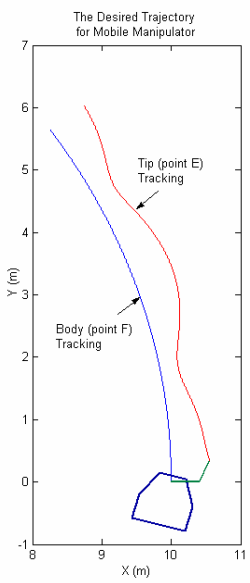


Figure 5.8: DDMR Trajectory

Figure 5.9: MM Trajectory

Table 5.3: The disturbances

Parameters	Mobile Platform /Vehicle only	Mobile Manipulator
Type of disturbances	1. Q-constant 2. Q-pulse 3. Q-vibration	1. Q-constant 2. Q-impacts 3. Q-vibration
Constant torques at wheels L/R	(5;-5) Nm	RAC only: 0, ±0.1, ±0.2 Nm RACAFC: ±2, ±5, ±20 Nm
Constant torques at link-1 &link-2		RAC only: 0, 0.1, 0.2 Nm RACAFC: 1, 5, 20 Nm
Frequencies of Q-pulse at (L;R) wheels	(1;1) Hz	
Amplitudes of Q-pulse at (L;R) wheels	(5;5) Nm	
Frequencies of Q-sine at (L;R) wheels	(5;5) Hz	0.06 m/s
Amplitudes of Q-pulse at (L;R) wheels	(5;5) Nm	(10;0)m
Impact Forces		See Figure 5.10.
Vibration Excitations		See Figure 5.11.

The impact disturbance was defined as a serially impact signals as shown in Figure 5.10 considering the robot operation at conditions with an instant collisions at the tip and holes or bumps at the terrain surface. We also applied vibration at each joint and wheel as depicted in Figure 5.11.

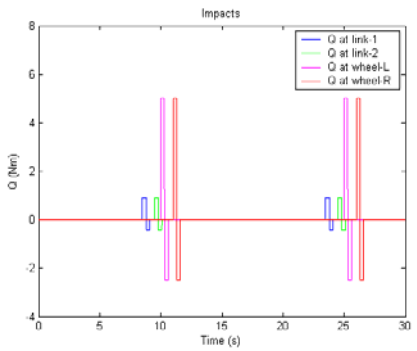


Figure 5.10: Impact Signals

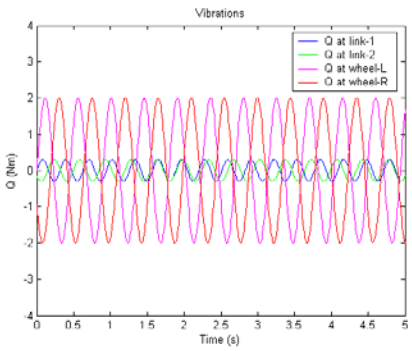


Figure 5.11: Vibration Signals

The frequencies and amplitudes of the impact forces and vibration excitations are shown in Table 5.4.

Table 5.4: Properties of impact and vibration disturbances

Parameters	Impact	Vibration
Frequency at link-1	15 Hz	20 Hz
Amplitude at link-1	(0.9;-0.45) Nm	0.3 Nm
Frequency at link-2	15 Hz	18 Hz
Amplitude at link-2	(0.9;-0.45) Nm	0.3 Nm
Frequency at wheel-L	15 Hz	14 Hz
Amplitude at wheel-L	(5;-2.5) Nm	2 Nm
Frequency at wheel-R	15 Hz	14 Hz
Amplitude at wheel-R	(5;-2.5) Nm	2 Nm

The simulation methods and its parameters are shown in Table 5.5.

Table 5.5: Simulation methods and parameters

Parameters	DDMR	Mobile Manipulator
Start Time	0 s	0 s
Stop Time	15 s	30 s
Solver	ODE45 (Dormand-Prince)	ODE45 (Dormand-Prince)
Stepping method	Variable-Step	
Max. step size	0.1	0.1
Min. step size	0.001	0.001
Initial step size	0.001	0.001
Relative tolerance	0.001	0.001
Absolute tolerance	auto	auto

5.4.2 Simulink Block Diagrams

The simulation consists of three AFC-based schemes, i.e., DDMR (Differentially Driven Mobile Robot)-RACAFC, MM-RAC, and MM-RACAFC. Figures 5.12, 5.13, and 5.14 depict the Simulink diagram of DDMR-RACAFC, MM-RAC, and MM-RACAFC respectively. For the DDMR-RACAFC, the scheme without AFC can be selected by clicking the *Selector* in Figure 5.12.

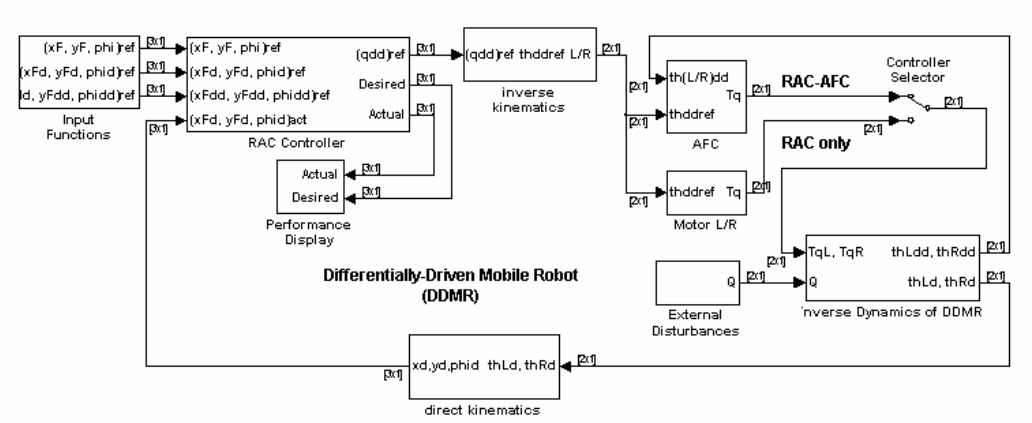


Figure 5.12: Simulink diagram of DDMR-RACAFC

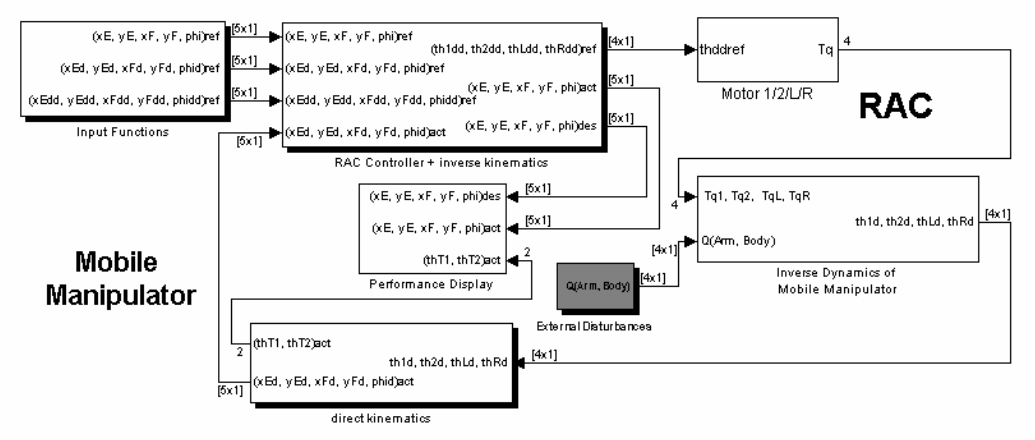


Figure 5.13: Simulink diagram of MM-RAC

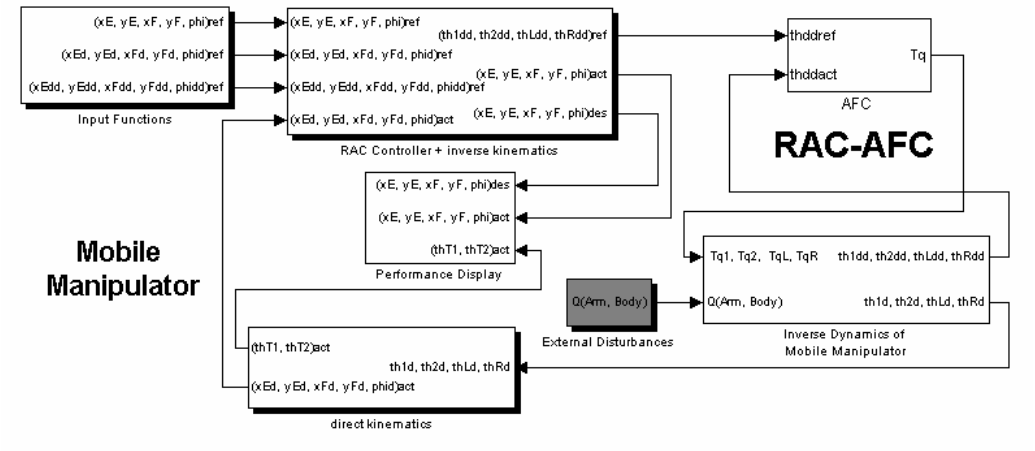


Figure 5.14: Simulink diagram of MM-RACAFC

Generally, these three schemes use same Simulink diagrams for certain functions in the blocks such as the input functions, RAC controller plus inverse kinematics, direct kinematics, performance display, inverse dynamics, and external disturbances. Each block in the diagram of DDMR (Figure 5.12) may be considered as part of the blocks that is related to the MM Simulink diagrams (Figures 5.13 or 5.14). The block of *Mobile Platform Input Function* in Figure 5.15 is used as the input of DDMR-RACAFC simulation. Its detailed schematic is shown in Figure 5.16.

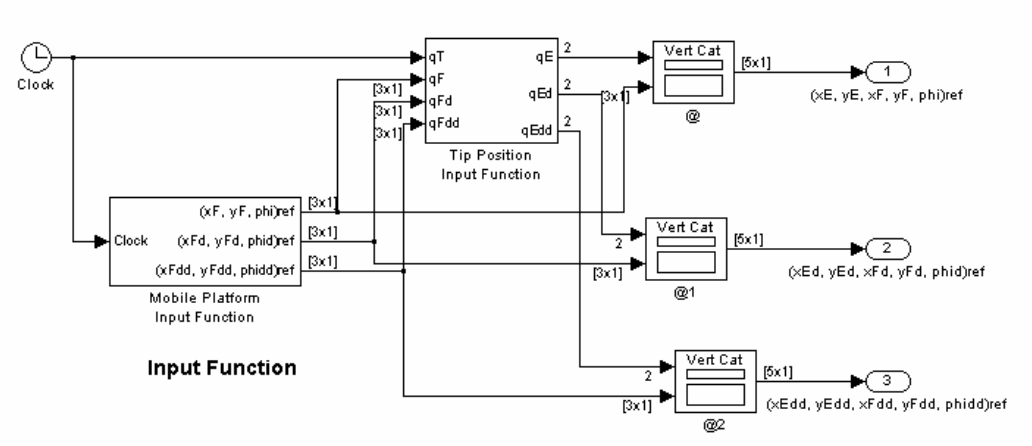


Figure 5.15: Simulink diagram of the input functions for MM simulation

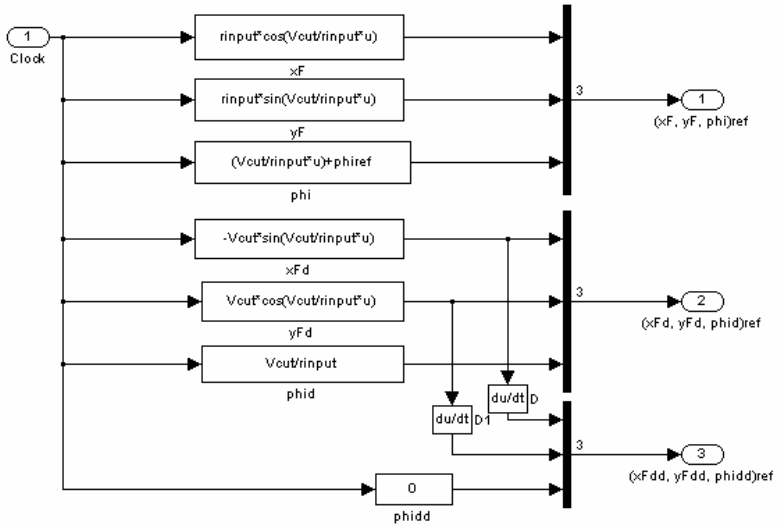


Figure 5.16: Simulink diagram of the *Mobile Platform Input Function*

Figure 5.17 shows the diagram of *Tip Position Input Function*. It is noted that the desired acceleration, whether for the mobile platform or the tip of arm, should be set according to span of the respective desired speed and position.

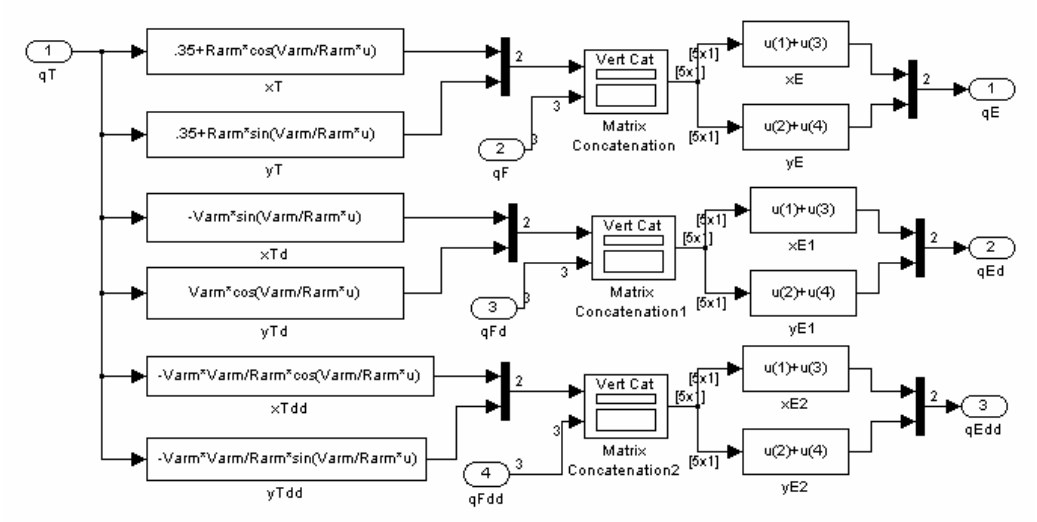


Figure 5.17: Simulink diagram of the *Tip Position Input Function*

Figure 5.18 shows the diagram of *RAC Controller* plus *Inverse Kinematics* section. The inverse kinematics is composed of inverse kinematics for the manipulator and for the platform. Figures 5.19 and 5.20 depict the Simulink diagrams of the respective blocks.

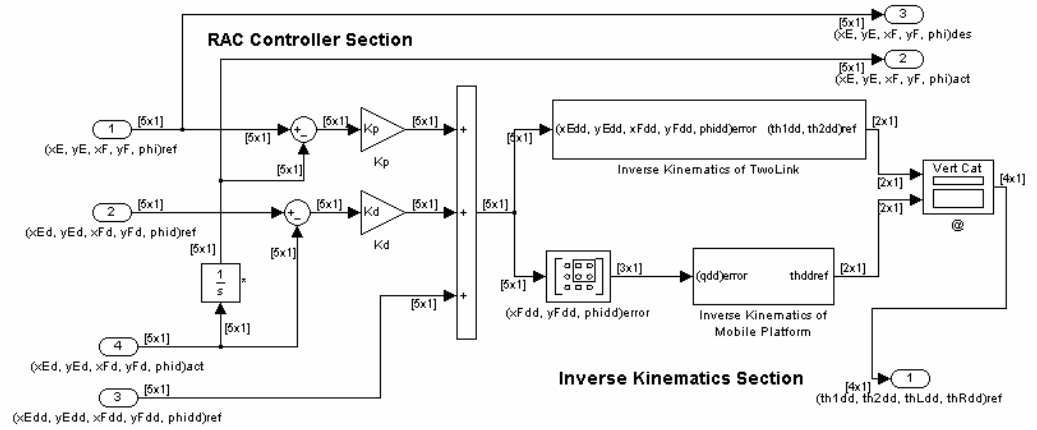


Figure 5.18: Simulink diagram of the *RAC Controller* plus *Inverse Kinematics* section

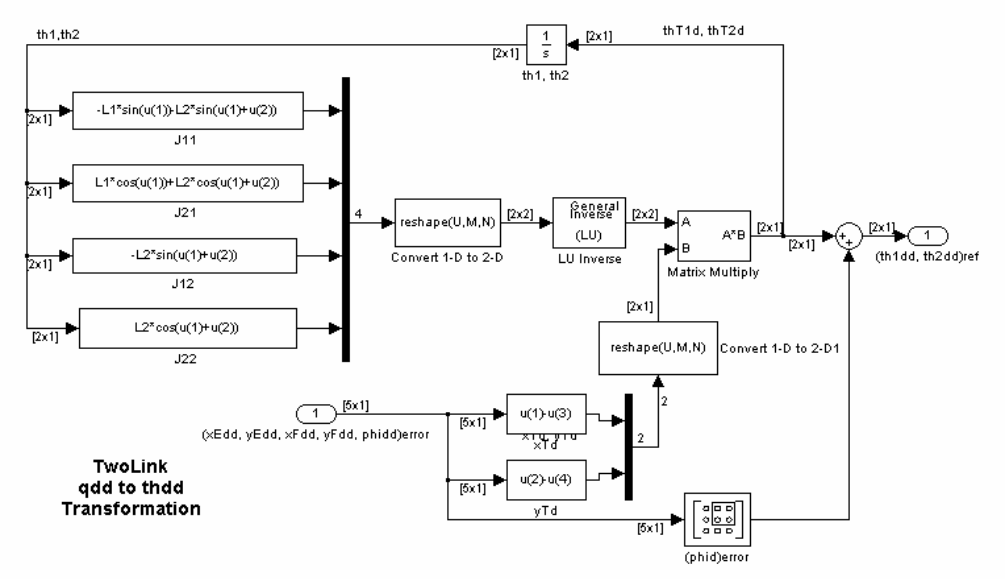


Figure 5.19: Simulink diagram of *Inverse Kinematics* of the manipulator

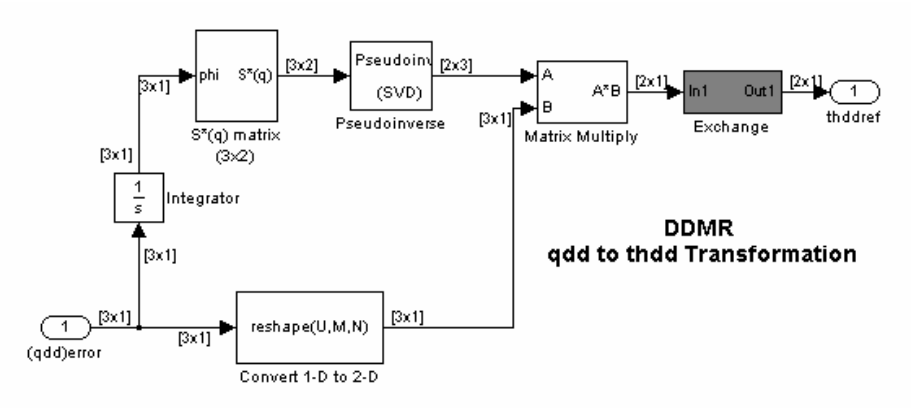


Figure 5.20: Simulink diagram of *Inverse Kinematics* of mobile platform

The diagram shown in Figure 5.20 is the inverse kinematics of mobile platform or DDMR. This block can be particularly used as the inverse kinematics section for DDMR simulation as in Figure 5.12. It is highlighted that transformation matrix $S^*(q)$ is as highlighted in Chapter 2.

The direct kinematics of the platform and the manipulator arm can be simply combined in one diagram as shown in Figure 5.21. Figures 5.22 and 5.23 depict the particular kinematics diagram for the respective platform and manipulator.

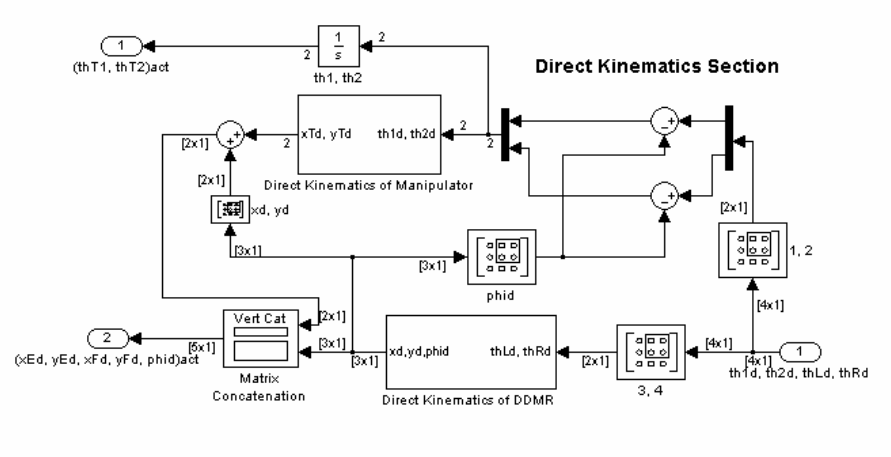


Figure 5.21: Simulink diagram of *Direct Kinematics* of MM

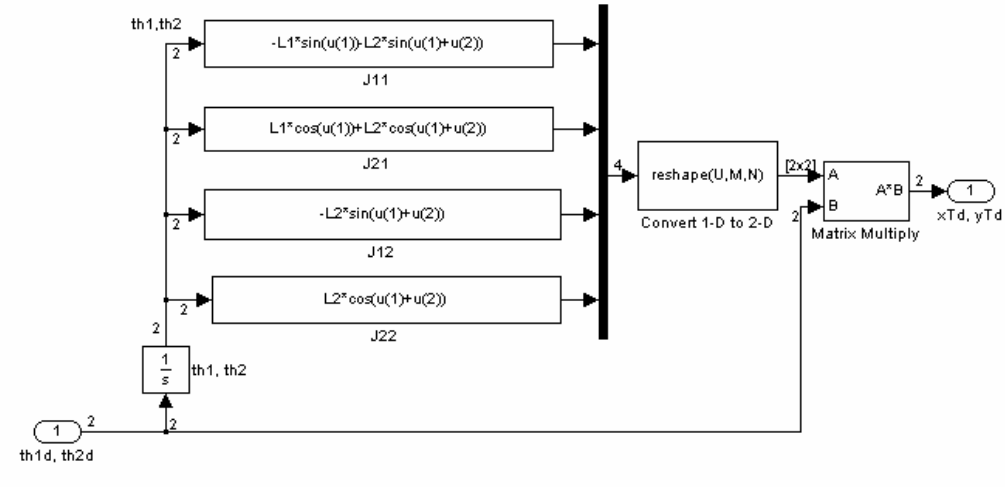


Figure 5.22: Simulink diagram of *Direct Kinematics* of manipulator

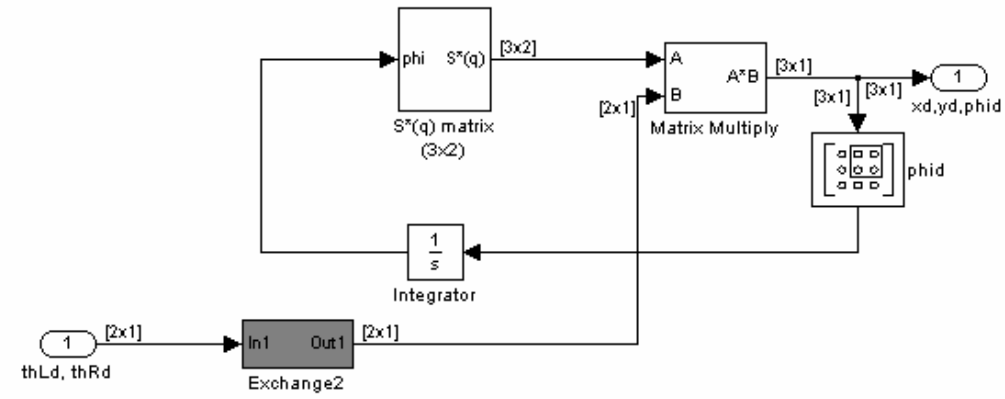


Figure 5.23: Simulink diagram of *Direct Kinematics* of mobile platform

For convenience, to analyze the simulation results in real-time, we applied an 'animation' procedure. Some of the important data of the robot operation such as the actual position of the platform (subject to point F) and of the tip (subject to point E) were also recorded in *Performance Display* section. Its diagram is shown in Figure 5.24.

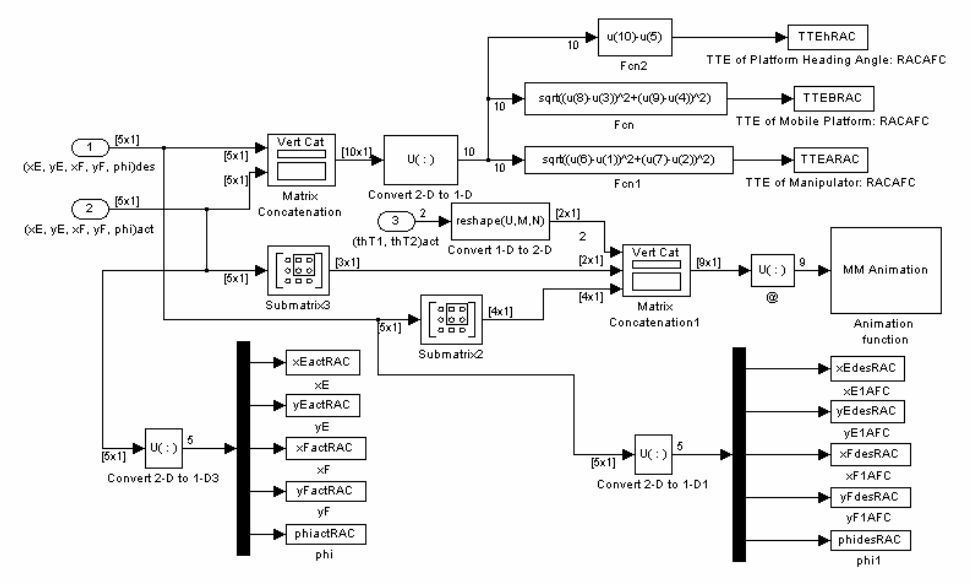


Figure 5.24: Simulink diagram of *Performance Display* section

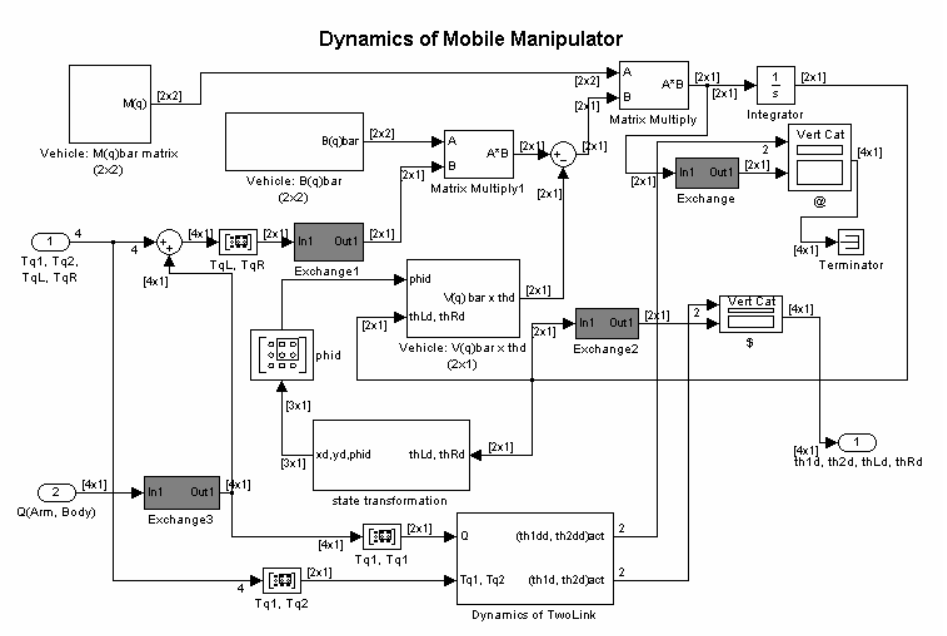


Figure 5.25: Simulink diagram of *Inverse Dynamics* of MM

The dynamics of the mobile manipulator consists of two parts, i.e., the DDMR and manipulator. In this study, we have neglected the interaction torques among the platform and the manipulator by considering the robot moving at relatively low speed, taking into account that the control problems of the platform and that of manipulator arm have been well understood (Kolmanovsky and McClamroch, 1995). Figure 5.25 shows a Simulink block diagram representing dynamics of mobile manipulator as mentioned earlier. Figures 5.26 and 5.27 are the dynamic constraint matrix, $M(q)$ and overall dynamics of the manipulator (arm) respectively.

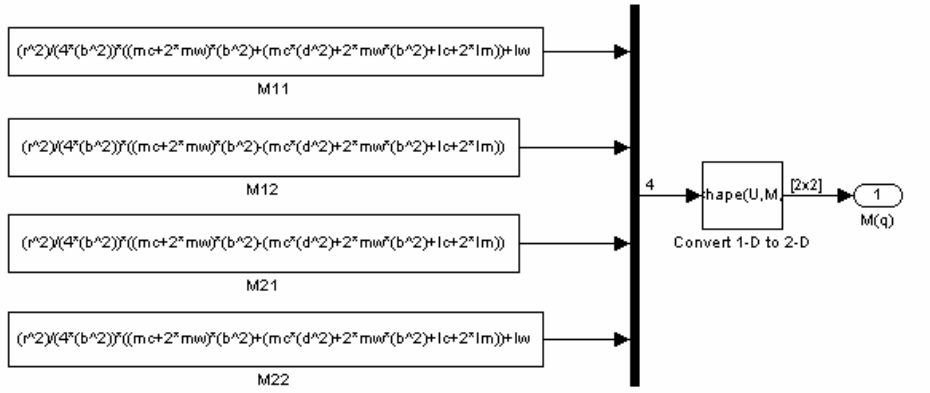


Figure 5.26: Simulink diagram of the constraint matrix $M(q)$

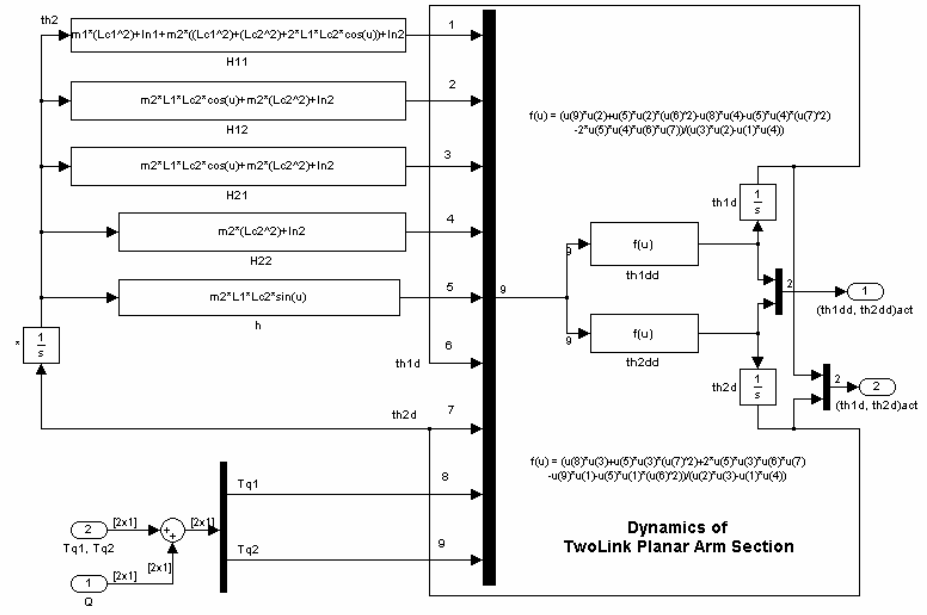


Figure 5.27: Simulink diagram of the manipulator's dynamics

5.5 Results and Discussion

The discussion of the simulation results obtained is divided into in two topics, i.e., based on DDMR simulation and MM simulation. For both simulations, RAC and or RACAFC schemes have been implemented. The analysis is concerned on the performances comparison between the two control concepts.

5.5.1 RACAFC for DDMR

The initial experiment is to determine appropriate *Kp* and *Kd* values of the RAC section (see Figure 5.12). The tuning process was performed in the RAC mode using a *trial-and-error* method. Some disturbances were also considered during the tuning process. By using the tuned *Kp* and *Kd* values, a number of experiments were performed to include the AFC scheme. The **IN** value was then tuned. It was obtained through a heuristic method. The range of the **IN** from 1 to 2.8 kgm^2 was optimized manually with a step of 0.1 kgm^2. The consideration was based on the minimum average of the tracking error's root mean square.

From the initial investigation on the RAC, the optimum *Kp* and *Kd* for $q = [x_F, y_F, \varphi]^T$ were obtained as follows:

$$Kp = \begin{bmatrix} 350 & 0 & 0 \\ 0 & 350 & 0 \\ 0 & 0 & 0.005 \end{bmatrix} \quad Kd = \begin{bmatrix} 320 & 0 & 0 \\ 0 & 320 & 0 \\ 0 & 0 & 0.0015 \end{bmatrix} \tag{5.20}$$

It should be noted that the *Kp* and *Kd* (0.005 and 0.0015 respectively) for the robot's heading angle control are relatively very small compared to those of the robot's movement control. This is due to the fact that the heading angle control is very sensitive in the proposed (x, y, φ) kinematics control mode. These control parameter values would be applied to the next investigation employing the AFC

scheme, initially to tune the *IN* value. Figure 5.28 shows the result of the **IN** tuning process.

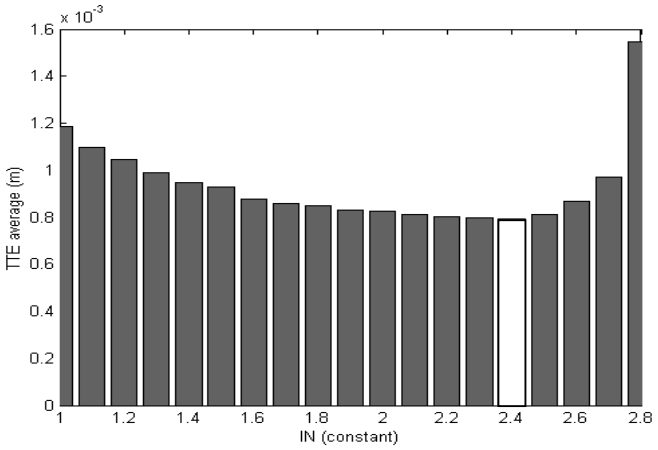
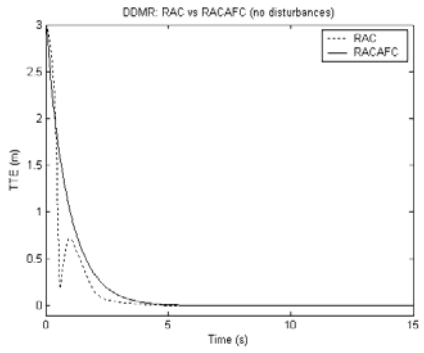


Figure 5.28: Results of **IN** tuning (the white bar indicates the optimum value)

From Figure 5.28, it can be deduced that the optimum **IN** is around 2.4 kgm^2. This value was then used as the reference **IN** for testing the robustness of the proposed DDMR-RACAFC scheme. **IN** can be written as:

$$IN = \begin{bmatrix} 2.4 & 0 \\ 0 & 2.4 \end{bmatrix} \tag{5.21}$$

Figure 5.29 shows the *Trajectory Tracking Error* (TTE) of the two schemes with and without disturbances. The disturbances were defined as a horizontal sinusoidal force applied to each wheel with amplitude of 5 N and frequency of 5 Hz as described in Table 5.3.



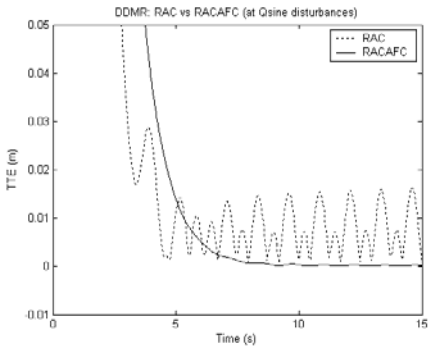


Figure 5.29 (a): TTE with no disturbances **Figure 5.29 (b):** TTE with disturbances

From Figure 5.29, it is obvious that the RACAFC method demonstrates its potential, particularly when the sinusoidal type disturbance is applied. The disturbance is significantly compensated and the errors generated were far less than shown by its counterpart. The average steady state error for the RAC method is around 20 mm in average but the RACAFC scheme successfully suppresses the error to less than 1 mm. The superiority of the AFC is easily concluded from the graphic 'animation' as shown in Figure 5.30.

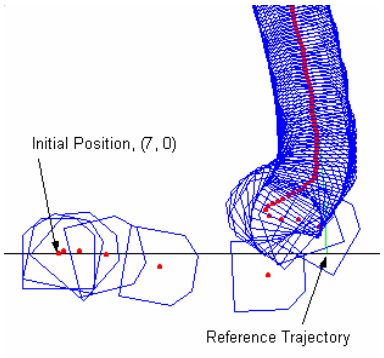


Figure 5.30 (a): Robot Animation for the RAC

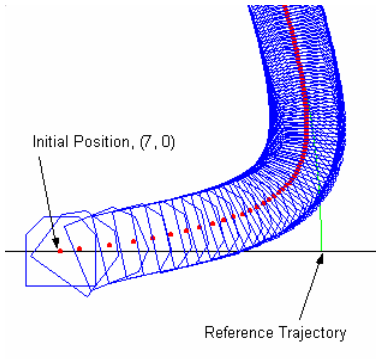


Figure 5.30 (b): Robot Animation for the RACAFC

For the robot overall dimension of 80×80 cm and the input task of curve tracking with a radius of 10 m, the track errors generated for both schemes, i.e., 2 cm and 0.1 cm are relatively not significant considering a regular mobile robot tracking. For a very high precision robot task planning and trajectory tracking, then the results based on accuracy are truly significant and meaningful. The result however highlights the great potential of the RACAFC scheme for this kind of application.

5.5.2 Mobile Manipulator Control

By considering the simulation results on DDMR system mentioned earlier and the results of the RACAFC applied to robot arm reported in Musa (1998) and Pitowarno (2002) the proposed MM-RACAFC could be conveniently performed.

Equations (5.8) to (5.12) and (5.13) to (5.17) can then be combined into an integrated motion control as well. As it is, there are five parts of RAC and corresponding five parts of AFC that were created and that the overall control scheme should be run simultaneously. The optimum **IN** of the mobile platform that has been obtained from previous investigation, i.e., $\mathbf{IN} = diag[2.4 \quad 2.4]$ was used for the mobile platform of the MM.

5.5.2.1 Simulation Procedure

The optimum *Kp*, *Kd* and **IN** that have been acquired in the RACAFC applied to DDMR were again used as the initial crude values to investigate the arm of the MM. The complete simulation procedure is depicted in Figure 5.31.

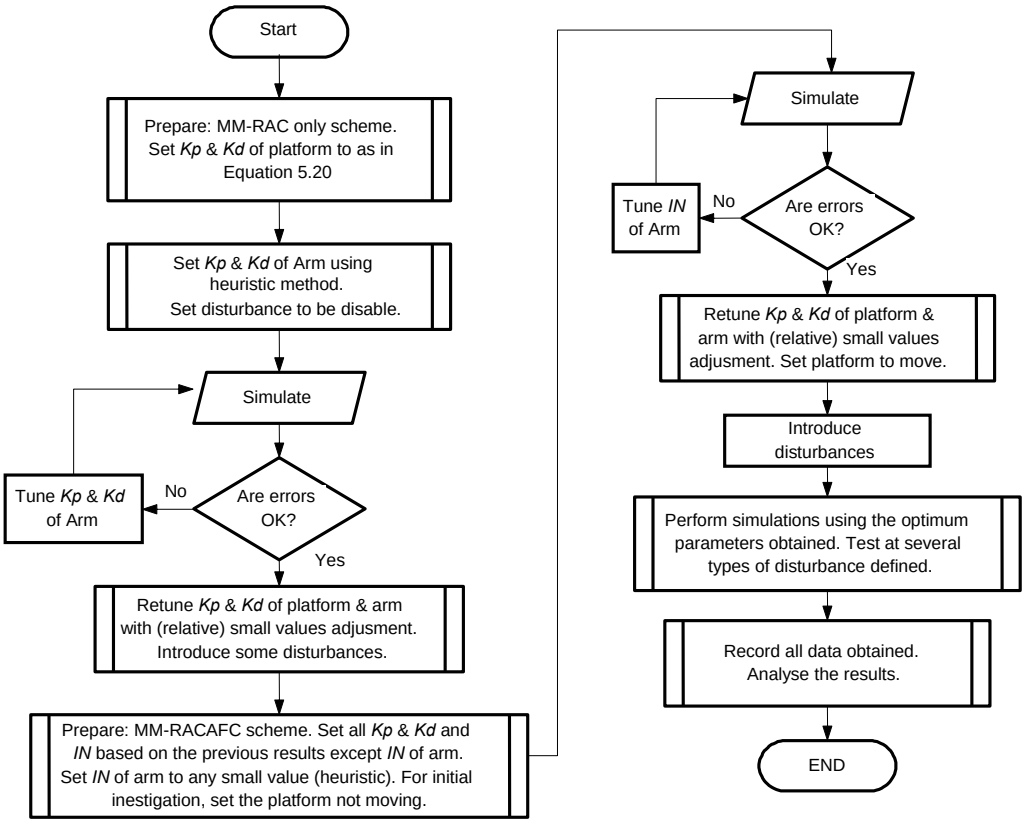


Figure 5.31: A flow chart showing the simulation procedure

The initial experiment was to determine the appropriate values of *Kp* and *Kd* of the RAC section. The tuning process was performed in the RAC mode using a heuristic method considering some disturbances in the process. By using the tuned *Kp* and *Kd* values, a number of experiments was then performed to include the AFC scheme. The *IN* value was also approximated in the same manner. For the mobile platform, the range of the *IN* value from 1 to 2.8 kgm^2 was optimized manually with a step of 0.1 kgm^2 as depicted in Figure 5.7. It was based on the minimum average of the tracking error's root mean square. For the arm or manipulator, its **IN** value was tuned by considering the platform not moving.

From Figure 5.31, it should be noted that the procedural step of retuning all the control parameters after the errors validation (at the left & right parts) using a very small values adjustment (up and down) was needed to refine the errors. In a heuristic or trial-and-error method this approach usually can improve the results although not always significant (Franklin, *et al*., 2002).

To clearly illustrate the potentials of the AFC that is incorporated into the RAC scheme, an analysis and discussion will be presented through a comparison study of the graphical results obtained. The discussion shall deal with the following headings:

- The optimum *Kp*, *Kd* and **IN** obtained.
- Effects of the constant torque disturbance.
- Effects of the impact disturbance.
- Effects of the vibration disturbance.

5.5.2.2 The Optimum *Kp, Kd* and IN obtained

The first procedure was to tune the *Kp* and *Kd* of the RAC using a heuristic method. Note that, in fact, the optimum *Kp* and *Kd* of the RAC was different with the

Kp and *Kd* of the RACAFC. *Kp* and *Kd* are expressed in diagonal matrices in the form:

$$Kp = diag\{Kp_{xE} \quad Kp_{xE} \quad Kp_{xF} \quad Kp_{yF} \quad Kp_{\varphi}\} \tag{5.22}$$

and *Kd* is:

$$Kd = diag\{Kd_{xE} \quad Kd_{xE} \quad Kd_{xF} \quad Kd_{yF} \quad Kd_{\varphi}\} \tag{5.23}$$

The results are shown in Equations (5.24) and (5.25). The RAC would be then evaluated using its optimum *Kp* and *Kd* as follows:

$$Kp_{RAC} = diag\{15 \quad 13.5 \quad 450 \quad 450 \quad 0.004\} \tag{5.24}$$

$$Kd_{RAC} = diag\{13 \quad 12 \quad 320 \quad 320 \quad 0.0017\} \tag{5.25}$$

Next, the optimum *Kp*, *Kd*, and *IN* of the RACAFC was investigated using the same method. The results are shown in Equations (5.26), (5.27) and (5.28) as follows:

$$Kp_{RACAFC} = diag\{350 \quad 350 \quad 450 \quad 450 \quad 0.004\} \tag{5.26}$$

$$Kd_{RACAFC} = diag\{320 \quad 320 \quad 320 \quad 320 \quad 0.0017\} \tag{5.27}$$

$$IN = \begin{bmatrix} IN_1 & 0 & 0 & 0 \\ 0 & IN_2 & 0 & 0 \\ 0 & 0 & IN_L & 0 \\ 0 & 0 & 0 & IN_R \end{bmatrix} = \begin{bmatrix} 0.0675 & 0 & 0 & 0 \\ 0 & 0.0675 & 0 & 0 \\ 0 & 0 & 2.4 & 0 \\ 0 & 0 & 0 & 2.4 \end{bmatrix} \tag{5.28}$$

It is noted that the values shown in Equations (5.26) and (5.27) are the refine-tuned *Kp* and *Kd* of the integrated kinematic control of the MM. The **IN** was then considered as the basic (crude) value for further experimentation.

5.5.2.3 Effects of Constant Torque Disturbance

Several types of constant torque (Qc) disturbances were introduced to the schemes. Qc is defined as:

$$Qc = \begin{bmatrix} Qc_1 & Qc_2 & Qc_L & Qc_R \end{bmatrix}^T \tag{5.29}$$

where Qc_1 is constant torque disturbance at joint-1, Qc_2 at joint-2, Qc_L at wheel-L, and Qc_R at wheel-R, and for the introduced disturbances we have used three types of Qc for the MM-RAC, and the other three types for MM-RACAFC, i.e.:

a. $Qc = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$ Nm (as no disturbances) for MM-RAC,

b. $Qc = \begin{bmatrix} 0.01 & 0.01 & 0.05 & -0.05 \end{bmatrix}^T$ Nm for MM-RAC,

c. $Qc = \begin{bmatrix} 0.02 & 0.02 & 0.1 & -0.1 \end{bmatrix}^T$ Nm for MM-RAC,

d. $Qc = \begin{bmatrix} 0.2 & 0.2 & 0.5 & -0.5 \end{bmatrix}^T$ Nm (as small Qc) for MM-RACAFC,

e. $Qc = \begin{bmatrix} 2 & 2 & 5 & -5 \end{bmatrix}^T$ Nm (as high Qc) for MM-RACAFC, and

f. $Qc = \begin{bmatrix} 20 & 20 & 20 & -20 \end{bmatrix}^T$ Nm (as very high Qc) for MM-RACAFC.

The different *Qc's* were applied to the two by considering the RACAFC was known very robust compared to the RAC only. Figure 5.32 depicts the error (TTE) at the tip end position for the MM-RAC and MM-RACAFC schemes.

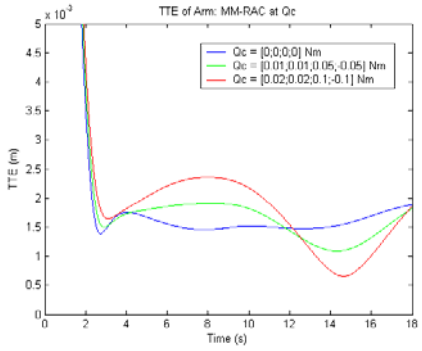


Figure 5.32 (a): TTE of Arm for MM-RAC at Qc

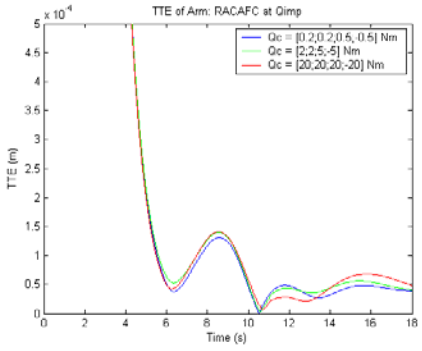


Figure 5.32 (b): TTE of Arm for MM-RACAFC at Qc

Figure 5.32 (a) clearly shows that the RAC has no ability to adapt if the operation condition is changed with the introduced disturbances. The increases of constant torques disturbance obviously degraded the RAC performance. At only *Qc* = [0.02;0.02;0.1;-0.1] Nm which relatively can be considered as the operation without disturbance (very small) the error has increased more than 1 mm peak-to-peak (indicated by the red line). In contrast, the RACAFC shows the robustness as depicted in Figure 5.32 (b). The applied disturbances at 0.5 to 20 Nm at the joints have been rejected by the AFC as well, and all the errors were compensated to less than 0.15 mm. Although the AFC is basically non adaptive scheme, but at this juncture, the term of active in active force control can be meant that the control is directly adaptable to the changes of the operation condition.

Figure 5.33 shows the generated errors of the two schemes at the body (subject to point *F* in Figure 2.1(b) of Chapter 2).

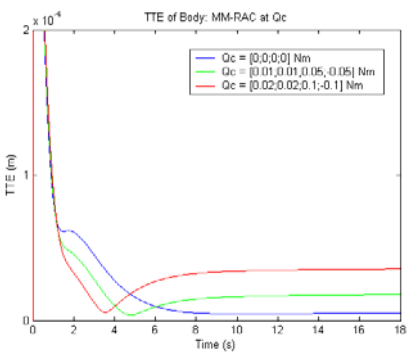


Figure 5.33 (a): TTE of Body for MM-RAC at *Qc*

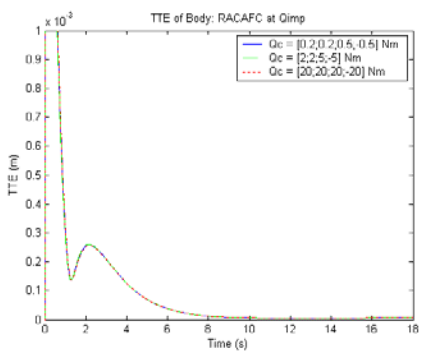


Figure 5.33 (b): TTE of Body for MM-RACAFC at *Qc*

The RACAFC also shows the robustness in the case of motion control with constant torques disturbance on the mobile platform as shown in Figure 5.33 (b). On the contrary, the performance of the RAC was gradually degraded when the disturbances was increased as shown in Figure 5.33 (a). At only a relative small disturbance, i.e., *Qc* = [0.02;0.02;0.1;-0.1] Nm at the joints and wheels, the error becomes three times bigger (indicated by the red line) than at the condition without disturbance (indicated by the blue line).

5.5.2.4 Effects of Impact Disturbance

Next, the proposed MM-RACAFC has been evaluated by introducing the impacts as depicted in Figure 5.10. We have used the term of Q_{gain} as a disturbance's gain factor that

$$Qimp = Q_{gain} \times Qimp_{\max} \tag{5.30}$$

where $Qimp$ is the applied impacts, and $Qimp_{\max}$ is the maximum impacts disturbance (Q_{gain} = 1). Q_{gain} is also used in the term of vibration (discussed later).

The impact disturbance (and the other defined disturbances) in the simulation model can be activated by selecting the parameters in a window of dialog box (by clicking the icon of the related block at the simulation diagram) as shown in Figure 5.34.

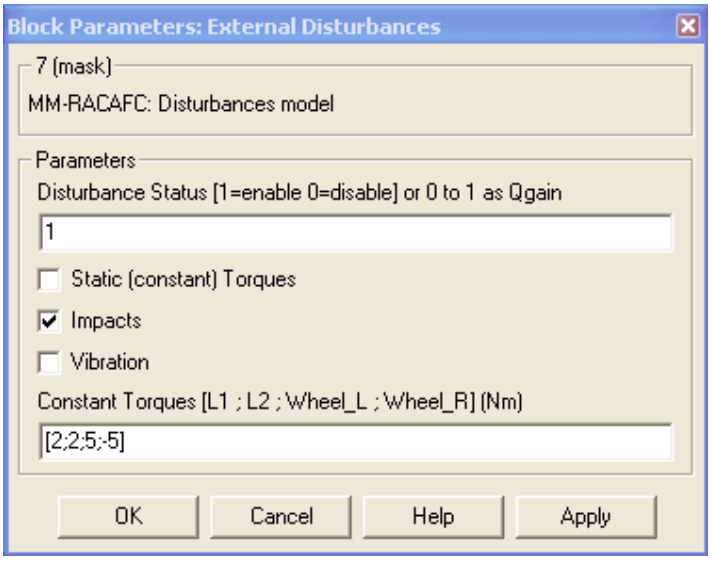


Figure 5.34: The dialog box for *External Disturbance Section*

The disturbance status (Q_{status}) in the window can be set to a value, 0 for disable, 1 for (full) enable, and 0< Q_{status} <1 to indicate the disturbance attenuation (or Q in the case of impacts). The impacts have been applied to the two schemes in

three conditions, i.e., small (Q_{gain} = 0.1), regular (Q_{gain} = 0.5), and high impact (Q_{gain} = 1). Figure 5.35 shows the response of MM-RAC and MM-RACAFC at the arm subjects to point E in Figure 2.1 (b) which the impacts is initially introduced at time of t = 8.5 s for the joints, and t = 10 s for the wheels. As can be seen, the RACAFC responds to the impacts disturbance much better that the RAC only does. The RAC, as aforementioned in the case of constant torques disturbance, could not adapt the change of condition of operation. Increasing Q_{gain} = 0.1 to Q_{gain} = 0.5 affects the error to increase to more than twice (less than 2 mm to 4 mm) as depicted in Figure 5.35 (a). On the contrary, the RACAFC remains stable and succeed to reject the impacts to less than 0.3 mm of the error at maximum impacts (Q_{gain} = 1).

It should be noted that the initial error of the RACAFC at t = 0 to 6 is the error occurred at the initial positioning of the tip regarding to the pose of the arm when the robot moving just started.

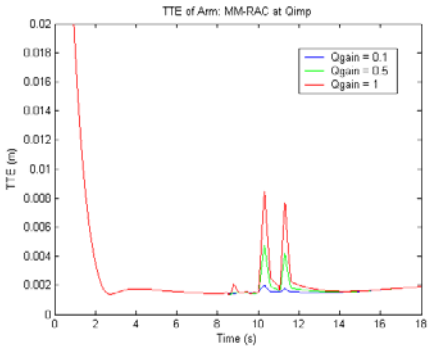


Figure 5.35 (a): TTE of Arm for MM-RAC at *Qimp*

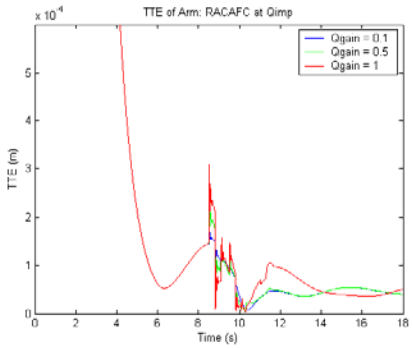


Figure 5.35 (b): TTE of Arm for MM-RACAFC at *Qimp*

A robust performance of the RACAFC is also shown in Figure 5.36 (b) in the case of the impacts occurred at the wheels of the body. The impacts with peak 5 Nm (See Figure 5.10) imposed to each the wheel has been rejected to less than 0.03 mm. In contrast, The RAC clearly shows the incapability on compensating the impacts disturbance as depicted in Figure 5.36 (a). The error is around ten times higher than the error on the RACAFC scheme.

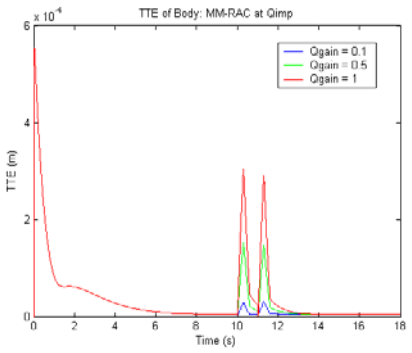


Figure 5.36 (a): TTE of Body for RAC at *Qimp*

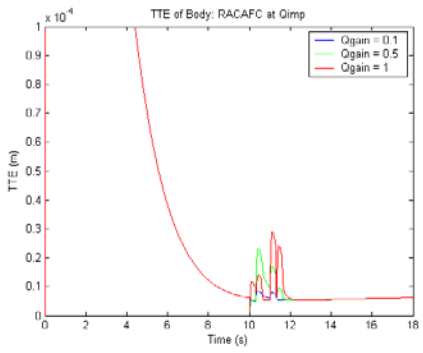


Figure 5.36 (b): TTE of Body for RACAFC at *Qimp*

It is noted that the errors of the arm in Figures 5.35 (a) and 5.35 (b) are the accumulated errors of the body and the arm itself because of the arm is mounted on the top of the body. As can be seen, when the body's error is increased at t = 11 s, as shown in Figure 5.36 (a), the error of the arm is also increased. Particularly, in the case of RAC only it can be deducted that RAC has no capability to regulate disturbance effects of the interaction among the body to the arm.

5.5.2.5 Effects of Vibration

Vibration typically occurs in real world robotic application. In a mobile system it maybe occurred at wheels on small waving surface of road or terrain where the robot moves. The vibration on wheels can serially affect to the joints of arm, moreover at the tip end position. In the simulation, vibrations at relative low frequencies were defined. With reference to Figure 5.11, the vibrations, *Qvib*, were defined to be 14 Hz at wheel-L and wheel –R with amplitude of 2 Nm, and 20 Hz at joint-1 and 18 Hz at joint-2 with amplitude of 0.3 Nm respectively. The vibrations were introduced to the robot operation during the trajectory tracking. Figure 5.37 shows the effects of the vibration for the MM. Again, the proposed RACAFC exposes the superiority in compensating the high non-linear disturbance such as the defined vibrations. The error of arm of the RACAFC is only less than 0.2 mm at the largest vibration (Q_{gain} = 1) while of the RAC is more than 3 mm. Although the applied AFC only used a basic (crude) approximation on estimating the inertia

matrix of the mass moment of the MM, the advantage of the proposed RACAFC is clear and meaningful. In addition, RAC and AFC can be simply considered as a general robust motion control of the mobile manipulator.

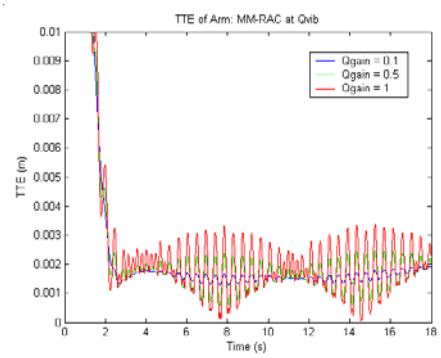


Figure 5.37 (a): TTE of Arm for MM-RAC at *Qvib*

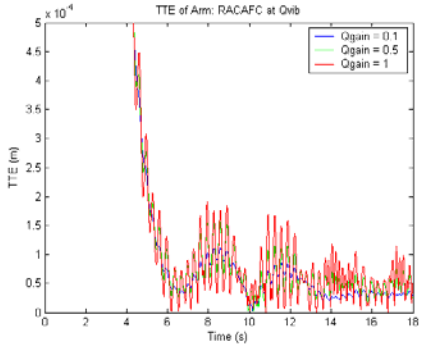


Figure 5.37 (b): TTE of Arm for MM-RACAFC at *Qvib*

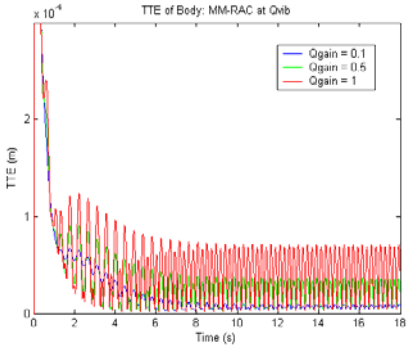


Figure 5.38 (a): TTE of Body for MM-RAC at *Qvib*

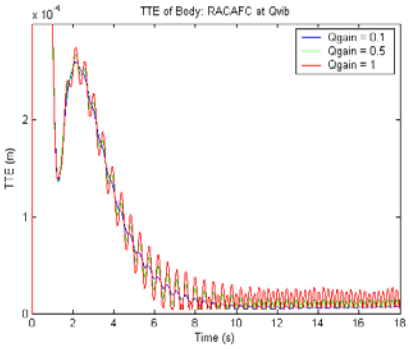


Figure 5.38 (b): TTE of Body for MM-RACAFC at *Qvib*

Figure 5.38 also shows the error refinement at the body subjects to point *G*. Though at the initial tracking (t = 0 to 5 s) the error is relatively high (because of the process to reach steady state), but however, the vibration was cancelled by the RACAFC to less than 0.03 mm, while the RAC has suppressed the vibration at 0.08 mm only.

5.6 Conclusion

Three major conclusions can be drawn from the study carried out in this chapter:

.

- The advantage of using the RACAFC scheme to perform the motion control of the DDMR with capabilities of compensating the effects of disturbances is demonstrated. The technique was found to be superior to using RAC only. The RAC only scheme is found to be feasibly applied to nonholonomic motion control of the system, thereby providing another alternative means of controlling the system other than the existing (v, ω) control. The RAC scheme augmented by combining the RAC with AFC strategy to the motion control of the system is considered as a totally new approach. However, a number of experimentation needs to be carried out to further explore the potential of the scheme when other different tasks, parameters or operating and loading conditions are considered.

- The proposed MM-RACAFC was successfully proven as a stable and robust motion control for mobile manipulator in this study. By using only the 'proportional' AFC thru optimizing the fixed **IN** the proposed scheme was capable to improve the RAC as a robust MM motion control.

- The proposed method successfully exhibits the system's robustness even in the presence of the introduced disturbances in the form of constant torque disturbance, impact and vibration forces.

CHAPTER 6

RESOLVED ACCELERATION CONTROL AND KNOWLEDGE-BASED FUZZY ACTIVE FORCE CONTROL (RACKBFAFC) OF MOBILE MANIPULATOR

6.1 Introduction

In this chapter, a simplified coordinate (x, y) and heading angle (φ) of a nonholonomic mobile manipulator motion control using RAC combined with a knowledge based fuzzy active force control (KBFAFC) strategy is presented. The scheme consists of two main parts that each has particular goal, that are the RAC deals with the kinematics, and the KBFAFC is designed to tackle the dynamics problem specifically. By using this RAC-based x and y control the proposed control scheme would have a more flexible position, velocity and acceleration control. This flexibility is gained by the use of simultaneous input reference position, velocity and acceleration parameters. An AFC scheme combined with the proposed knowledge-based fuzzy system is incorporated into the RAC scheme to tackle the robot's dynamic problem particularly those involving disturbances and uncertainties. The stability and robustness of the proposed RACKBFAFC are investigated through a simulation study. Several types of disturbances such as static torques, impacts and vibration are introduced to evaluate the performances. A deliberate analysis would then be discussed. In the last subsection, the investigation has showed the significant improvement of the robustness of the proposed RACKBFAFC compared to the existing RACAFC.

6.2 Proposed MM-RACKBFAFC Scheme

The proposed scheme as shown in Figure 6.1 is made up of two controllers, namely, RAC and KBFAFC that can be theoretically designed independently.

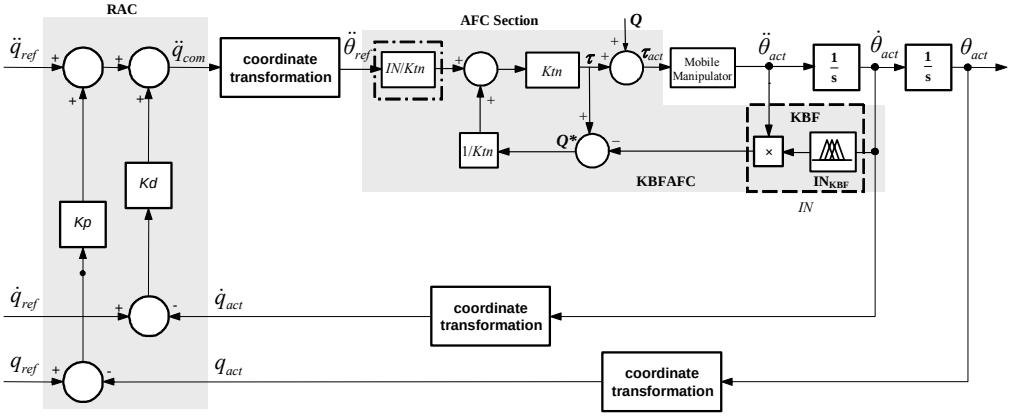


Figure 6.1: The proposed MM-RACKBFAFC

In Figure 6.1, the subscripts of *ref*, *com* and *act* denote the *reference*, *command* and *actual* respectively. Q indicates the bounded (known or unknown) disturbances, K_{tn} are the torque constants for the actuating motors (left and right wheels of the platform and joint-1 and joint-2 of the manipulator), $\mathbf{IN_{KBF}}$ is the estimated inertia matrix from the KBF system, and Q* is the estimated disturbance calculated by the KBFAFC algorithm. The *IN*′ on the left side is the estimated **IN** that is based on the optimum **IN** obtained from the RACAFC scheme. It is noted that the input of the $\mathbf{IN_{KBF}}$ function is the velocities at the joints and wheels.

6.2.1 RAC Section

The RAC applied to the RACKBFAFC is exactly the same as in the previous RACAFC scheme which has been discussed and established in Chapter 5. RAC is specifically designed to handle the entire kinematic problem of the MM while the

scheme of KBFAFC that is incorporated serially to the RAC deals with the dynamic aspect.

For convenience, the important RAC output equations are rewritten in this section as follows:

$$\ddot{x}_{Fcom} = \ddot{x}_{F\,ref} + Kd(\dot{x}_{F\,ref} - \dot{x}_{F\,act}) + Kp(x_{F\,ref} - x_{F\,act}) \tag{6.1}$$

$$\ddot{y}_{Fcom} = \ddot{y}_{F\,ref} + Kd(\dot{y}_{F\,ref} - \dot{y}_{F\,act}) + Kp(y_{F\,ref} - y_{F\,act}) \tag{6.2}$$

$$\ddot{\varphi}_{Fcom} = \ddot{\varphi}_{ref} + Kd(\dot{\varphi}_{ref} - \dot{\varphi}_{act}) + Kp(\varphi_{ref} - \varphi_{act}) \tag{6.3}$$

$$\ddot{x}_{Ecom} = \ddot{x}_{E\,ref} + Kd(\dot{x}_{E\,ref} - \dot{x}_{E\,act}) + Kp(x_{E\,ref} - x_{E\,act}) \tag{6.4}$$

$$\ddot{y}_{Ecom} = \ddot{y}_{E\,ref} + Kd(\dot{y}_{E\,ref} - \dot{y}_{E\,act}) + Kp(y_{E\,ref} - y_{E\,act}) \tag{6.5}$$

The subscripts *ref*, *com*, *act* and *e* refer to the input reference, command, actual output and error respectively. *Kp* and *Kd* are the proportional and derivative gains respectively. All controller output equations in Equations (6.1) to (6.5) have negative feedback elements that contribute to the generation of relevant error signals that are subsequently coupled with the respective controller gains. In the global MM motion control, the controller equations can be considered as separated controls but to be executed simultaneously in real-time.

6.2.2 KBFAFC Section

The KBFAFC is principally an AFC-based scheme in which the **IN** is intelligently computed via a knowledge-based fuzzy (KBF) method. Recalling the output equation of the AFC algorithm in Equations (5.6) and (5.7) and referring to Figure 6.1, we have:

$$\tau = IN'\left(\ddot{\theta}_{ref} - \ddot{\theta}'_{act}\right) + I'_m Ktn \tag{6.6}$$

or alternatively, it can be rewritten as:

$$\tau = IN'\ddot{\theta}_{ref} - IN'\ddot{\theta}'_{act} + I'_m Ktn \tag{6.7}$$

In Figure 6.1, the adaptation through the KBF algorithm is imposed on the $\mathbf{IN_{KBF}}$. Thus, the KBFAFC output equation can be written as:

$$\tau = IN_F\ddot{\theta}_{ref} - IN_{KBF}\ddot{\theta}'_{act} + I'_m Ktn \tag{6.8}$$

and the simplified dynamic equation of the system is given by:

$$\tau_{act} = IN_F\ddot{\theta}_{ref} - IN_{KBF}\ddot{\theta}'_{act} + I'_m Ktn + Q \tag{6.9}$$

with

$$IN_{KBF} = f_{KBF}(\dot{\theta}_{act}) \tag{6.10}$$

where IN_F is the fixed **IN** that was obtained as the optimum **IN** of the existing RACAFC (established in Chapter 5), and $\mathbf{IN}_{KBF}$ is the variable or adaptive **IN** that should be computed by the KBF algorithm (f_{KBF}) based on the actual velocity $\dot{\theta}_{act}$. The superscript $'$ denotes an estimated or measured quantity. The angular velocity to be controlled (for each joint and wheel) is employed as the respective input function of f_{KBF}. For KBF design, the knowledge-based reasoning as the basis for the development of the fuzzy inference mechanism should take into consideration the knowledge investigation and representation, and knowledge acquisition and processing. These will be duly presented and discussed in the following subsections.

6.2.3 Knowledge Investigation and Representation

The first procedure to implement the knowledge based fuzzy system is knowledge investigation. The main aim of this procedure is to ensure sufficient and appropriate knowledge are gathered for the purpose of reasoning in the process of the building the fuzzy system that will be applied to the AFC. In a standard knowledge-

based system, the knowledge can be anything (information/data) that is captured from the system. In a global view point, the knowledge on mobile manipulator system and its environments include task specifications, control system characteristics and its responses and the global constraints (kinematics and dynamics) of the robot world as can be illustrated in Figure 6.2.

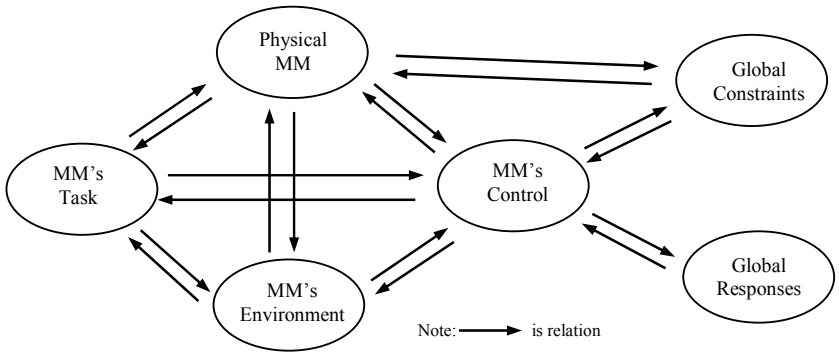


Figure 6.2: An illustration of global knowledge of mobile manipulator

In principle, we can start investigating knowledge from any node in Figure 6.2. The arrows are used to relate the knowledge being captured to a more concern view of knowledge in the next relation. In this study we have designed an investigation procedure that has the direction of 'flow' as depicted in Figure 6.3.

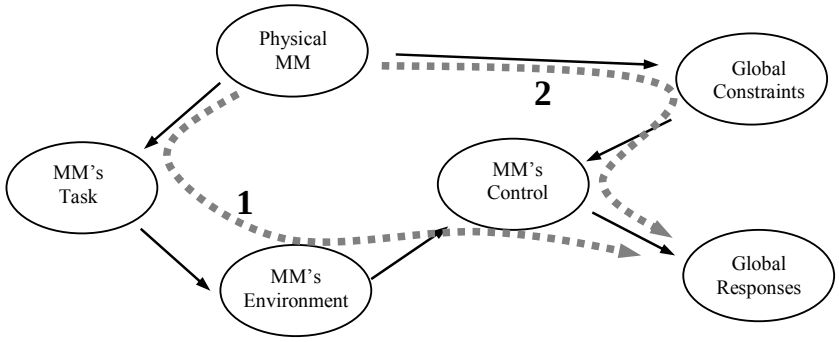


Figure 6.3: The selected semantic networks of the MM's knowledge structure

Figure 6.3 basically consists of two standard semantic networks. The first network follows a flow that serially configured as steps of relation of *Physical MM*

→ *MM's Task* → *MM's Environment* → *MM's Control* → *Global Responses*. The second flow is *Physical MM* → *Global Constraints* → *MM's Control* → *Global Responses*. At this juncture, the two procedural knowledge investigations are performed with the same most concerned views that are global responses. The node *MM's Control* is used as an overlapping view point that can be more qualitatively investigated. Following similar methods as those executed by Kuwata and Yatsu (1997) and Hildebrand and Fathi (2004), the view point of the node *MM's Control* can be expressed as:

$$Vi = \{Nc_{MM's\_Control}, \{\cup^{\vee} R\}, STEP\} \tag{6.11}$$

where *Vi* is the view point, *Nc* is the current node, $\{\cup^{\vee} R\}$ is a set of labels and *STEP* is a qualitative investigation. In this study, we have selected a qualitative investigation that includes the entire previous AFC-based control schemes, the robot's tasks and the effects of disturbances (with or without disturbances). Figure 6.4 shows the qualitative investigation in a semantic network that has been adopted in the study.

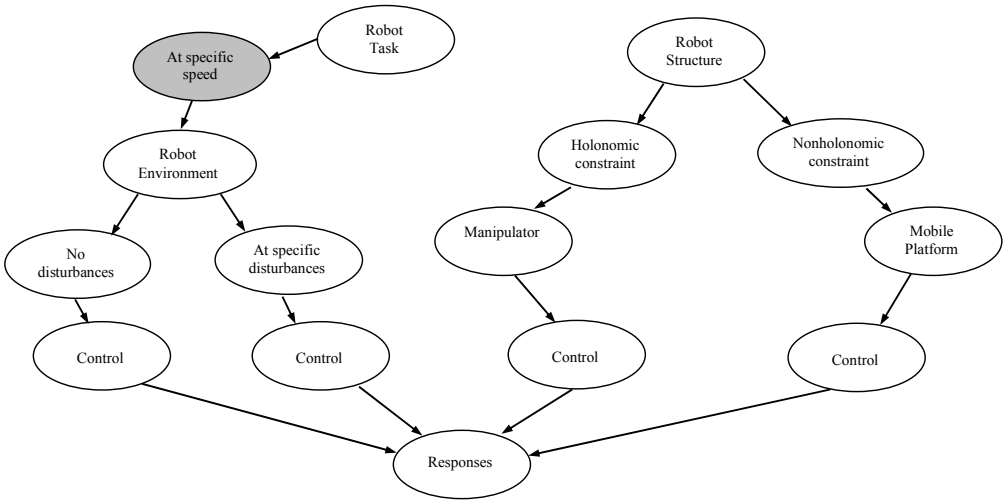


Figure 6.4: The qualitative investigation in a semantic network

The nodes of responses in Figure 6.4 are used as qualitative measurements in which each of the prior node can be inferred to them. Some specified input functions are then implemented in both simulation and the experimental works so that the mobile manipulator can be programmed to perform specific trajectory tracking task. The common control scheme used for both studies is the RACAFC using fixed inertia matrix estimator. Note that the scheme is applied to both studies prior to the development of the proposed RAC-KBFAFC scheme. Once the knowledge is fully acquired from the knowledge investigation procedure, it is then implemented into the design of the proposed scheme. Illustrations of *Features* or *Knowledge Extraction Procedure* in *Knowledge Investigation* are shown in Figures 6.5 (a) to 6.5(c). The figures illustrate how the knowledge required by the proposed system is investigated.

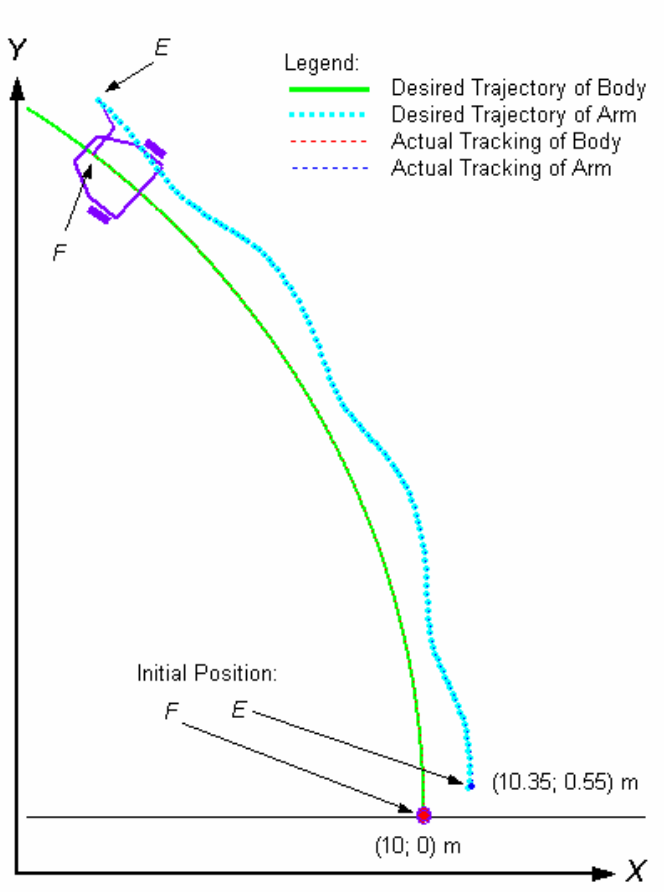


Figure 6.5 (a): The trajectory tracking of the mobile manipulator

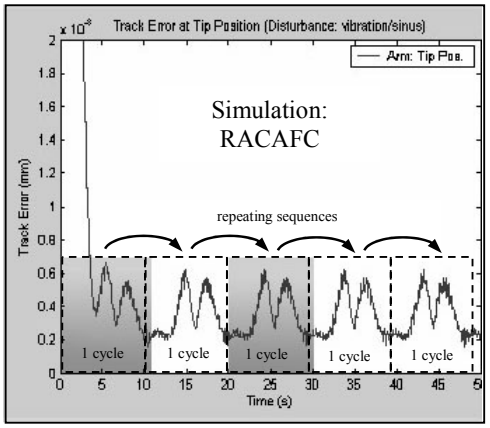


Figure 6.5 (b): The track error at the tip end position for five cycles of repeating tasks in the simulation

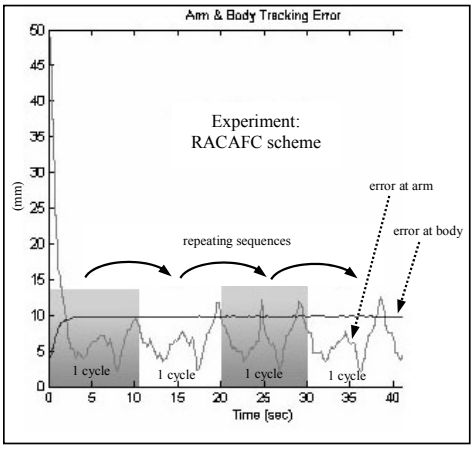


Figure 6.5 (c): The track error at the tip end position for four cycles of repeating tasks in the experiment

Figure 6.5 (a) depicts the movement of the mobile manipulator in which it can be seen that the manipulator's tip position produces a curvy path in the direction of the robot's forward movement. Figure 6.5 (b) shows the error signal produced from the simulation work while Figure 6.5 (c) exhibits the corresponding error curve based from the experimental study. It is obvious that the error patterns for both studies are repetitive in nature. Also, it is found that the same input function that is used in the simulation study which when implemented to the experimental robot, yields similar error pattern over a period of time. Thus, some features of the track error signal patterns have been investigated and identified. This information provides the required 'knowledge' for the subsequent implementation of the overall control scheme. It is evident that the error changes (increases or decreases) continuously according to specific positions of the robot system during the tracking operation.

Based on the knowledge investigation procedure, some hypotheses can be deduced as follows:

- There exists a correlation between the track error pattern and the non linearity of the system dynamics specifically related to the robot motion. In this context, the

inertia matrix **IN** of the manipulator can be considered as the non-linear element of interest.

- The ripples, spikes, and other features of the error pattern show the 'intensity' of the non linearity of the system that occur at specific locations and times.

- Some repeating error signal patterns produced at certain robot motion event could specifically imply the 'knowledge' that could provide a means to improve the control method. Hence, it is thought to be a wise move such that the effective robot control action should not take into account the robot motion moving along the full trajectory but consider only at locations where 'positive' definite error patterns occurred.

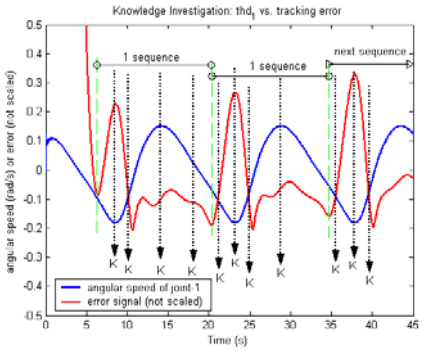


Figure 6.6 (a): Angular velocity versus track error at joint-1

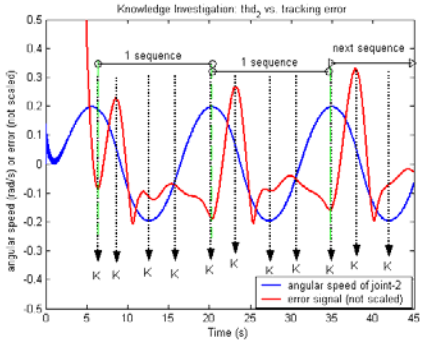


Figure 6.6 (b): Angular velocity versus track error at joint-2

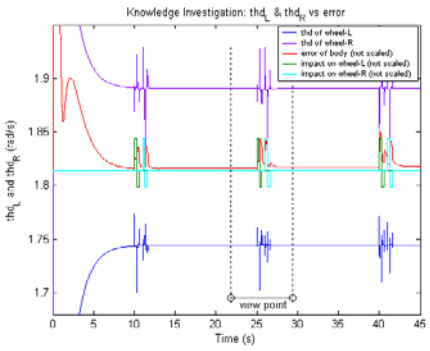


Figure 6.6 (c): Angular velocity versus track error at wheel-L

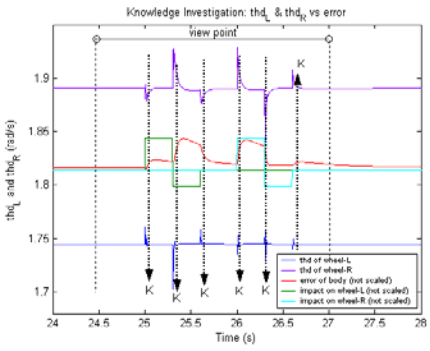


Figure 6.6 (d): Angular velocity versus track error at wheel-R at a view ponit

The response visualized in the respective error signal pattern of the certain robot operation at the certain condition is then analyzed as shown in Figures 6.6 (a) to 6.6 (d). In the figures, the notation *K* denotes the knowledge that is intensively investigated at particular time instants. From Figures 6.6 (a) and 6.6 (b), it can be seen that the instant error signal patterns at times denoted by K are unique and exhibits relationships with the concurrent actual velocity at the joints. It is also occurred at the wheels as can be seen in Figures 6.6 (c) and 6.6 (d). When the velocity is changed subjects to the impact disturbance, the error signal pattern on body is specific. Based on this we then concluded that each of the response can inherently exhibits the 'behavior' of the control, or in short we call the behavior as the captured knowledge. Table 6.1 shows the representation of knowledge that had been captured from the investigation.

Table 6.1: The knowledge representation

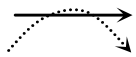
Prior knowledge	**Related knowledge**	**Relation descriptions**	**Specific Responses**	**Empirical hypothesis/ suggestion**
Robot's subsystems (joints & wheels)	Partial control (holonomic or nonholonomic)	Each subsystem is controlled partially	Error in each subsystem is unique	Error in each subsystem can be refined partially
Inertia of each subsystem	Uncertainties (of the environment & task)	The uncertainties affect each subsystem specifically	Error in each subsystem is unique	Each of the estimated inertia can be refined specifically
Robot's specific task	Actual (angular) velocity of each subsystem	Specific task implies specific actual velocity of each subsystem	**Case 1**: When actual velocity is going to smaller at top curvy velocity, the error is going to larger. **Case2**: When actual velocity is going to larger at bottom curvy velocity, the error is going to smaller. **Case 3**: When actual velocity is going to *smaller*, the error is found *large*. **Case 4**: When actual velocity is going to *larger*, the error is also found *small*.	- The increased error exhibits that the actual inertia of subsystem is increased or decreased. - It is therefore the estimated inertia value should be increased or decreased that follows "behavior" of the actual velocity.

From Table 6.1, the inertia matrix estimation or adaptation can be deduced that the increased error can be inferred as the condition that the actual estimated inertia matrix applied to the AFC is going to mismatch. Thus, the applied estimated inertia matrix should adaptively be increased or decreased regarding the amount of error. The decision to increase or decrease the applied estimated inertia matrix depends on the vector (or direction) of the actual velocity.

6.2.4 Knowledge Acquisition and Processing

From the hypotheses deduced, an inference mechanism can be derived. In this study, the angular velocity of the robot arm is chosen as the indicator (input function) to describe the relationship between the produced error and the actual velocity. The output variable to be computed is the estimated inertia matrix, **IN** of the system that is necessary for AFC mechanism. Table 1 shows the derived inference mechanism.

Table 6.2: The inference mechanism

Case	Part of velocity signal pattern	Related part of error signal pattern	Actions to be taken
1			set IN pattern as top-bottom-top direction
2			set IN pattern as top-bottom-top direction
3			set IN pattern as bottom-top-bottom direction
4			set IN pattern as bottom-top-bottom direction
Note: The basic IN (top value) uses the crude's $IN = diag\{0.0675\ 0.0675\ 2.4\ 2.4\}\text{kgm}^2$			

The inference mechanism given in Table 6.2 can be illustrated visually by studying the graphics shown in Figures 6.7 (a) to 6.7 (d) after the knowledge has

been taken into account and the output variable has been constructed based on the input function. The figures illustrate the relationship of velocity and inertia matrix that subject the joints and wheels.

Figures 6.7 (a) and 6.7 (b) are with the inferred input-output relation of knowledge at the joints, while Figures 6.7 (c) and 6.7 (d) are for the wheels. In Figures 6.7 (b) and 6.7 (d), the red dot-line indicate the fixed **IN** without adaptation (established in the RACAFC scheme). By incorporating the appropriate relationship between the input and the output variables, the design of the fuzzy system is clear. Hence, a simple fuzzy structure is selected to demonstrate the proposed method that can in turn lead to simple adjustment of the fuzzy specification.

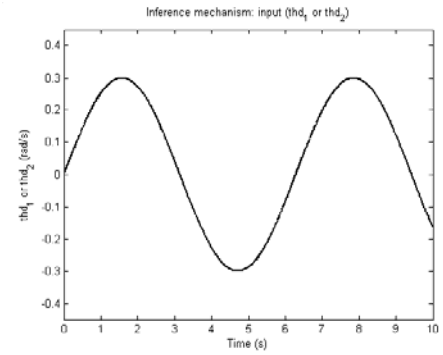


Figure 6.7 (a): Inference input signal of KBF for manipulator

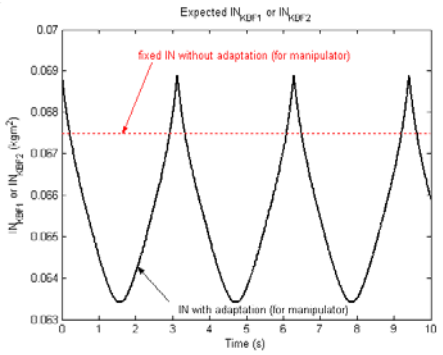


Figure 6.7 (b): Expected output signal of KBF for manipulator

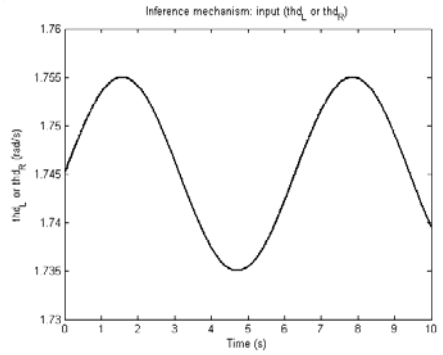


Figure 6.7 (c): Inference input signal of KBF for mobile platform

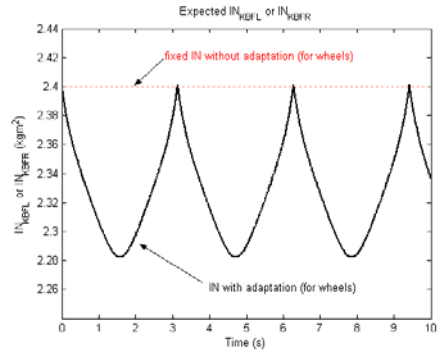


Figure 6.7 (d): Expected output signal of KBF for mobile platform

6.2.5 KBF Design

From the procedure of the above knowledge-based method and considering Tables 6.1 and 6.2 a set of KBF function can be derived as follows:

$$IN_{KBF1} = f_{KBF1}(\dot{\theta}_{act1}) \tag{6.12}$$

$$IN_{KBF2} = f_{KBF2}(\dot{\theta}_{act2}) \tag{6.13}$$

$$IN_{KBFL} = f_{KBFL}(\dot{\theta}_{actL}) \tag{6.14}$$

$$IN_{KBFR} = f_{KBFR}(\dot{\theta}_{actR}) \tag{6.15}$$

where $f_{KBF1}(\dot{\theta}_{act1})$, $f_{KBF2}(\dot{\theta}_{act2})$, $f_{KBFL}(\dot{\theta}_{actL})$, and $f_{KBFR}(\dot{\theta}_{actR})$ are specific for the respective joints or wheels. In this study we have considered to design the KBF for each of subsystem partially in simply form of fuzzy system with single input and single output rather than designing a universal KBF with four inputs ($\dot{\theta}_{act1}$, $\dot{\theta}_{act2}$, $\dot{\theta}_{actL}$, and $\dot{\theta}_{actR}$) and four outputs (IN_{KBF1}, IN_{KBF2}, IN_{KBFL}, and IN_{KBFR}). The advantage of the consideration is the designed partial fuzzy system can be simple and easy to maintain.

Considering the obvious goal of the fuzzy system design as mentioned above the proposed RACKBFAFC scheme is then designed as simple as possible. Hence, a *Mamdani* type fuzzy with three triangular shaped membership functions (MFs) representing the input and three *Gaussian* shaped membership functions of the output are selected for each input-output relationship. The MFs of the inputs are namely *smaller*, *normal* and *larger*. We used the term *smaller* instead of SMALL to emphasize that *smaller* is a derived knowledge that more meaningful rather than SMALL which is merely an information. The MF is typically constructed to represent the middle (or average) of the input value. Also, the term of *larger* can be more meaningful than LARGE with respect to knowledge-based system. It is to distinguish the meaning of information and knowledge more precisely (Marakas, 1999).

For the manipulator, *normal* is set to *zero* velocity by considering that the joints move in two directions (clock wise and counter-clock wise) with limited angle (not more than 180˚ to left or right), and in the operation, *zero* is the condition in which the link or joint is not actually moved. For the platform, *normal* is selected to be 1.815 rad/s to meet with desired linear velocity of the MM subjects to point *F*. However, the setting of *normal* is adjusted manually with a relative small value by considering the respective desired angular velocity of joints or wheels. Figures 6.8 (a) to (d) show the implementation of the inference mechanism in the KBF design.

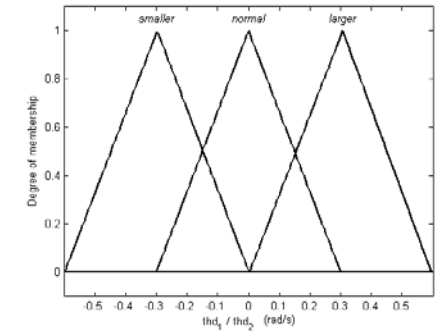


Figure 6.8 (a): MFs of input of joint-1 and joint-2

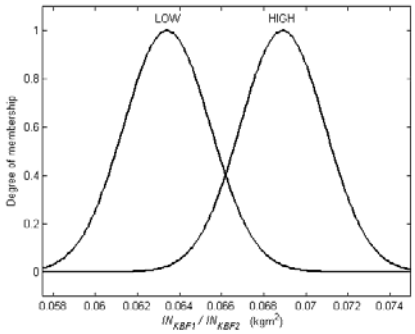


Figure 6.8 (b): MFs of output of joint-1 and joint-2

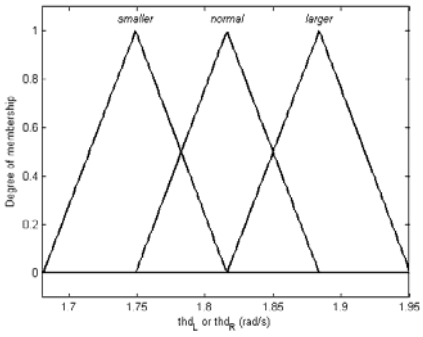


Figure 6.8 (c): MFs of input of wheel-L and wheel-R

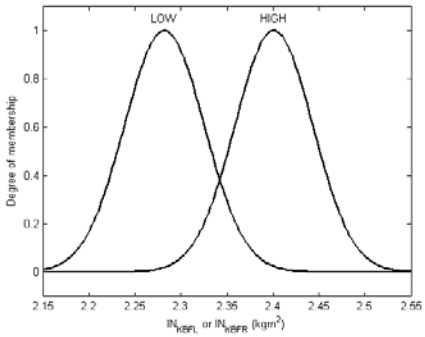


Figure 6.8 (d): MFs of output of wheel-L and wheel-R

By considering Figures 6.8 (a) to 6.8 (d) and regarding to Figures 6.7 (a) to 6.7 (d), all the respective KBF system design can be simply done using basic *Mamdani* types of fuzzy logic with some considerations as follows:

- AND method: *minimum*,

- OR method: *maximum*,
- Implication method: *minimum*,
- Aggregation method: *maximum*, and
- Defuzzyfication method: *centroid*.

The rules that satisfy the inference mechanism shown in Figures 6.7 (a) to 6.7 (d) are then simply designed as follows:

1. **IF** *actual velocity is smaller* **THEN** *inertia matrix is LOW*
2. **IF** *actual velocity is normal* **THEN** *output is HIGH*
3. **IF** *actual velocity is larger* **THEN** *output is LOW*

From the off-line test, this fuzzy system can easily perform the input-output mapping. The fuzzy system is then embedded into the overall control scheme for on-line implementation.

6.3 Simulation

Figure 6.9 shows a Simulink model of the MM-RACKBFAFC.

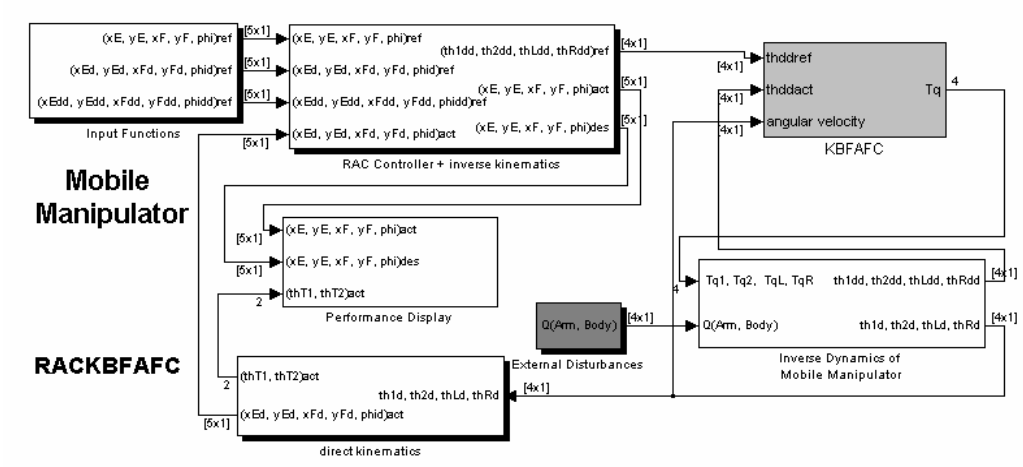


Figure 6.9: Simulink diagram of the proposed MM-RACKBFAFC

All of the sub diagrams of blocks in Figure 6.9, that are *Input Functions*, *RAC Controller* plus *Inverse Kinematics*, *Direct Kinematics*, *Performance Display*, *Inverse Dynamics of Mobile Manipulator*, and *External Disturbances* are generally same with the diagrams of the previous RACAFC scheme which has been established in Chapters 5 except for the block KBFAFC. Figure 6.10 (a) depicts the Simulink diagram contained in the KBFAFC block while Figure 6.10 (b) shows the detailed diagram of the KBF system.

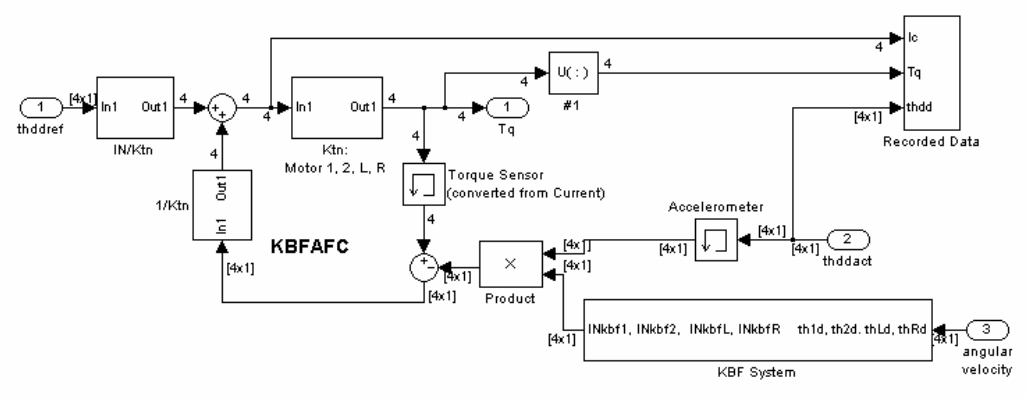


Figure 6.10 (a): Simulink diagram of the block of KBFAFC

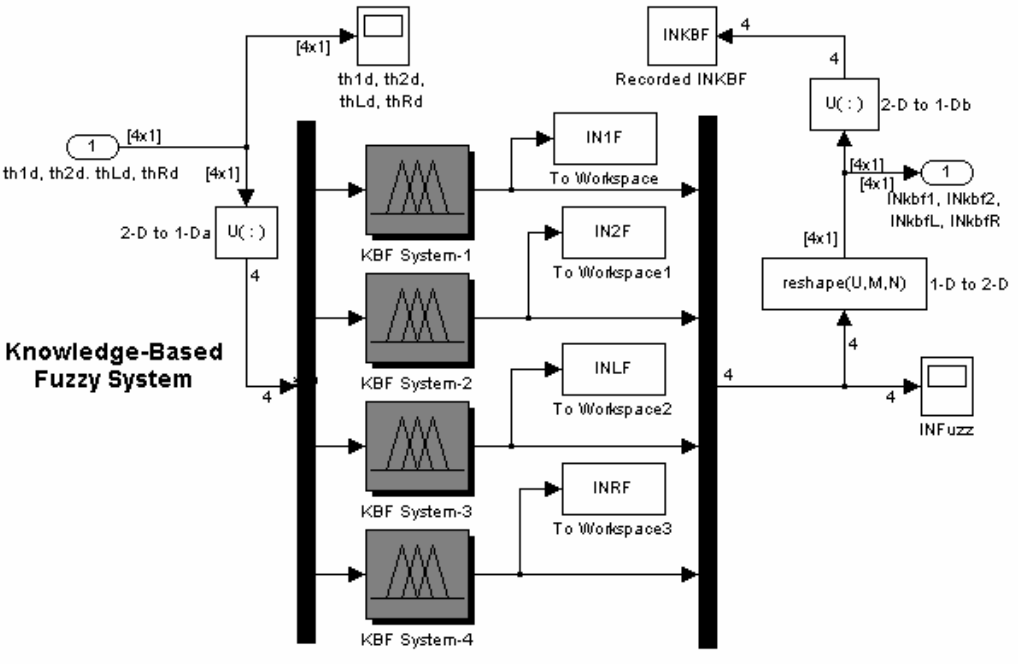


Figure 6.10 (b): Simulink diagram of the block of KBF System

The given task of the MM is to move its platform in a circular motion with a curvature radius of 10 m, at a velocity of 0.2 m/s (at point F) and the initial heading angle orientation of π/2.4 rad to the zero angle of the world *Cartesian* coordinate. The manipulator is programmed to follow a specified curve track at the right-hand side of the platform starting from (10.41, 0.35) m of the world *Cartesian* coordinate. The initial tip position is set to point (10.55, 0.35) m.

The values of *Kp* and *Kd* of the RAC for the AFC schemes have been obtained previously, that are $Kp_{RACAFC} = diag\{350 \quad 350 \quad 450 \quad 450 \quad 0.004\}$ and $Kd_{RACAFC} = diag\{320 \quad 320 \quad 320 \quad 320 \quad 0.0017\}$. These values are consequently used in the RAC part for all the proposed schemes to ensure a fair comparison can be made.

6.4 Results and Discussion

All the results obtained from the RACKBFAFC scheme would be compared to those of RACAFC. The discussion deals with the following headings:

- Effects of Constant Torque Disturbance, *Qc*
- Effects of Impact Disturbance, *Qimp*
- Effects of Vibration, *Qvib*

6.4.1 Effects of Constant Torque Disturbance

As an initial evaluation on the robustness, some constant torque disturbances, *Qc*, are introduced to the RACKBFAFC where $Qc = [Qc_{link\text{-}1}; Qc_{link\text{-}2}; Qc_{wheel\text{-}L}; Qc_{wheel\text{-}R}]$ is set as follows:

a. $Qc = \begin{bmatrix} 2 & 2 & 5 & -5 \end{bmatrix}^T$ Nm (high *Qc*),

b. $Qc = \begin{bmatrix} 30 & 30 & 30 & -30 \end{bmatrix}^T$ Nm (super high *Qc*).

Figures 6.11 (a) to 6.10 (d) show TTE of the arm and body of the RACKBFAFC versus RACAFC scheme. From the figures, it is obvious that RACKBFAFC is superior to RACAFC. The change of constant torques disturbance can be significantly compensated by the KBFAFC even the super-high *Qc*.

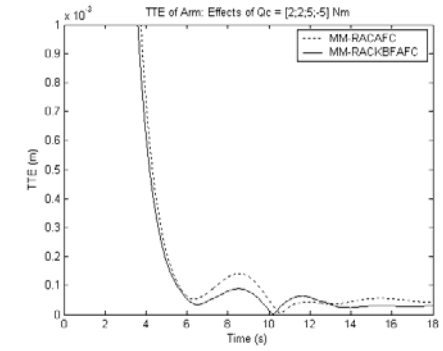


Figure 6.11 (a): TTE of Arm, AFC/ KBFAFC, *Qc* = [0.2;0.2;0.5;-0.5] Nm

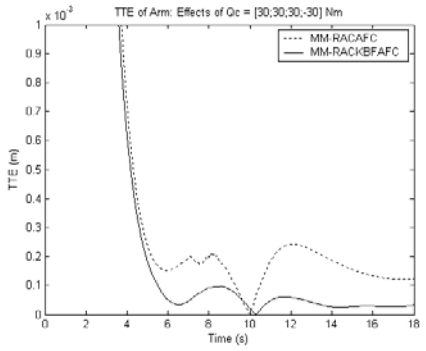


Figure 6.11 (b): TTE of Arm, AFC/ KBFAFC, *Qc* = [2;2;5;-5] Nm

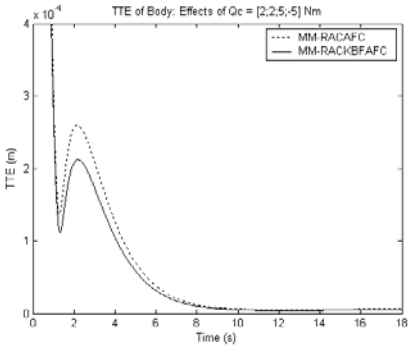


Figure 6.11 (c): TTE of Body, AFC/ KBFAFC, *Qc* = [20;20;20;-20] Nm

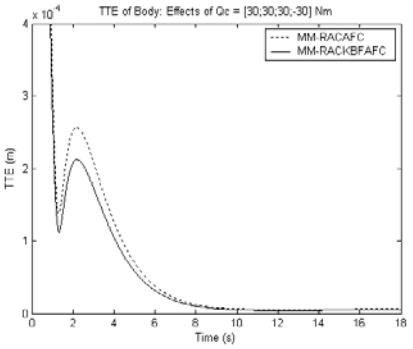


Figure 6.11 (d): TTE of Arm, AFC/ KBFAFC, *Qc* = [30;30;30;-30] Nm

6.4.2 Effects of Impact Disturbance

The impact disturbances as in the preceding chapters with additional term of Q_{gain} as gain are introduced to test the other criteria of robustness of the proposed scheme. Figures 6.12 (a) and 6.12 (b) show the response at the tip end position for MM-RACAFC and MM-RACKBFAFC which the impacts are introduced at t = 6.5 s for the joints, and t = 10 s for the wheels. As can be seen, the error on RACAFC has

tendency to be larger subjects to the increase of amplitude of impacts. In contrast, the RACKBFAFC persistently rejects the effect of impacts even at the highest impact (Q_{gain} = 3). In this case, it can be stated that in responses, the RACKBFAFC is similarly an adaptive scheme which has persistent robustness.

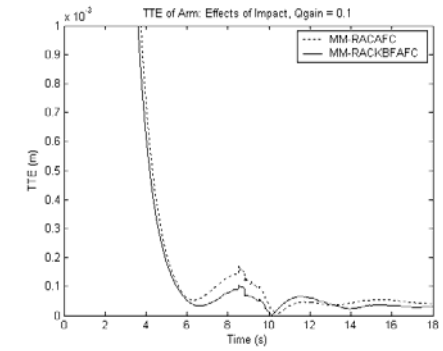


Figure 6.12 (a): TTE of Arm, AFC/ KBFAFC, *Qimp,* Q_{gain} = 0.1

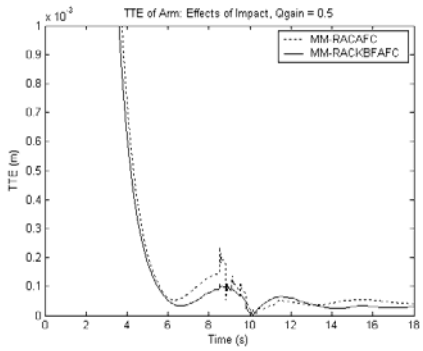


Figure 6.12 (b): TTE of Arm, AFC/ KBFAFC, *Qimp,* Q_{gain} = 0.5

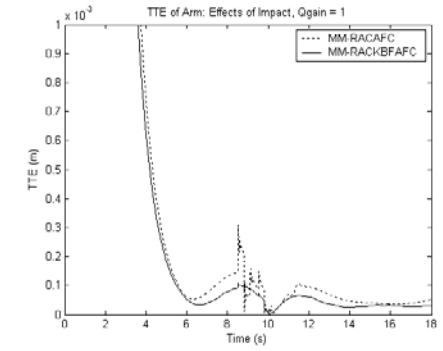


Figure 6.12 (c): TTE of Arm, AFC/ KBFAFC, *Qimp,* Q_{gain} = 1

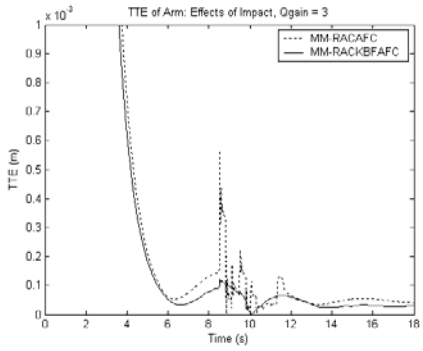


Figure 6.12 (d): TTE of Arm, AFC/ KBFAFC, *Qimp,* Q_{gain} = 3

Assuming the road surface of robot's workspace has uncertainties such as holes or bumps then the evaluation on the body deals with the impacts type of disturbance is also performed. Figures 6.13 (a) to 6.13 (d) show the error at point *F* of the body due to the impacts with Q_{gain} = 0.1, 0.5, 1.0, and 3.0.

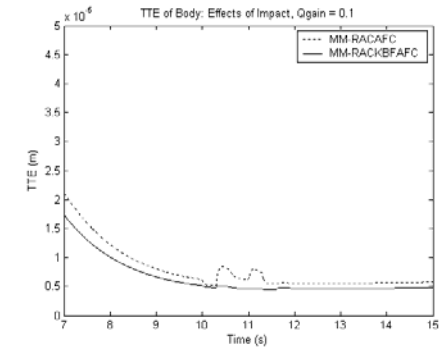


Figure 6.13 (a): TTE of Body, AFC/ KBFAFC, *Qimp,* Q_{gain} = 0.1

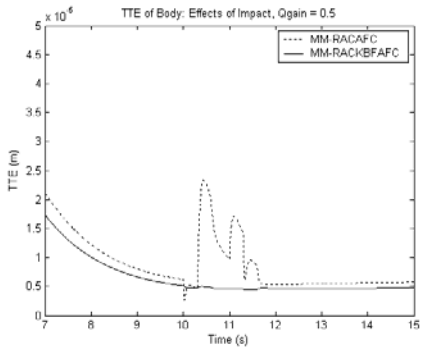


Figure 6.13 (b): TTE of Body, AFC/ KBFAFC, *Qimp,* Q_{gain} = 0.5

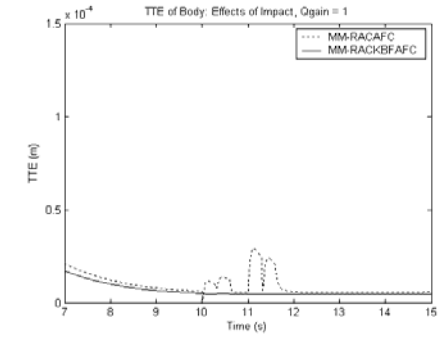


Figure 6.13 (c): TTE of Body, AFC/ KBFAFC, *Qimp,* Q_{gain} = 1

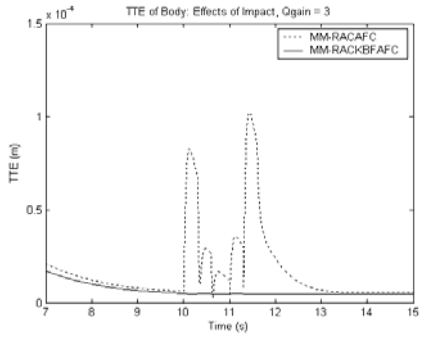


Figure 6.13 (d): TTE of Body, AFC/ KBFAFC, *Qimp,* Q_{gain} = 3

The results obtained shown in the figures are very clear. The RACKBFAFC applied to the platform can totally reject the impacts, even at Q_{gain} = 3. The method to adapt the inertia matrix estimation based on evidences (disturbance existences that measured from the actual velocity) is proven to be very effective.

6.4.3 Effects of Vibration

The robustness of the proposed scheme is also tested by introducing the vibration along the trajectory tracking. Figures 6.14 (a) to 6.14 (d) show the effects of the vibration on the MM-RACKBFAFC at the tip end position.

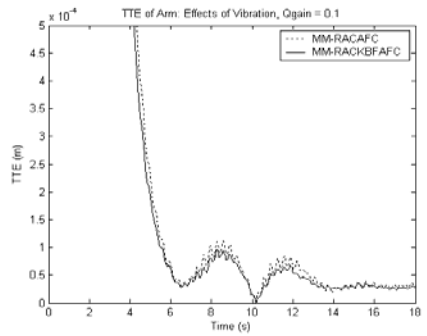


Figure 6.14 (a): TTE of Arm, AFC/ KBFAFC, *Qvib,* Q_{gain} = 0.1

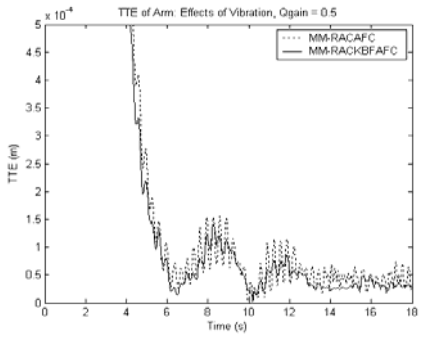


Figure 6.14 (b): TTE of Arm, AFC/ KBFAFC, *Qvib,* Q_{gain} = 0.5

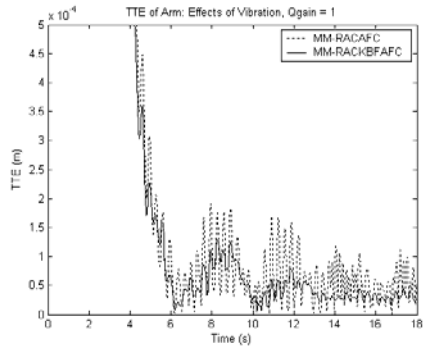


Figure 6.14 (c): TTE of Arm, AFC/ KBFAFC, *Qvib,* Q_{gain} = 1

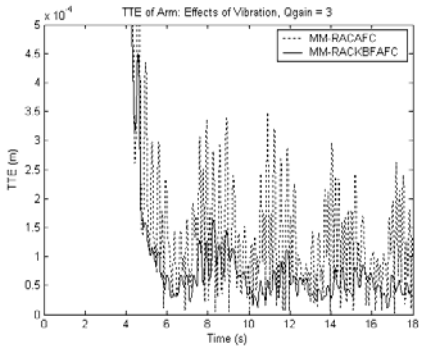


Figure 6.14 (d): TTE of Arm, AFC/ KBFAFC, *Qvib,* Q_{gain} = 3

As with the impact disturbances, the proposed RACKBFAFC also shows its superiority in rejecting the vibration at the tip end position. Even at $Q_{\text{gain}} = 3$, the error is still less than 0.15 mm. The same applies for $Q_{\text{gain}} = 0.5$ and 1.

The effect of vibration on the body is also well compensated by the RACKBFAFC as shown in Figures 6.15 (a) to 6.15 (d). As can be seen, the steady-state error of body for all the introduced vibrations is less than 0.03 mm in average. Specifically, from Figure 6.15 (d), it is obvious that the RACKBFAFC performs much better than RACAFC in which the error is shown to reduce considerably.

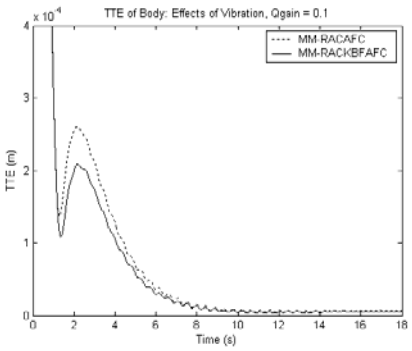


Figure 6.15 (a): TTE of Body, AFC/ KBFAFC, *Qvib,* Q_{gain} = 0.1

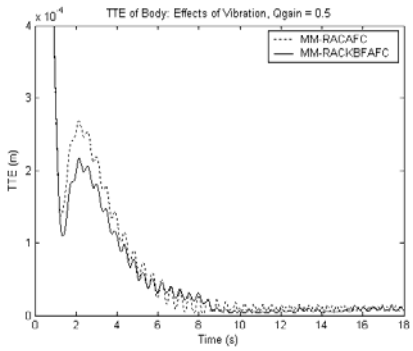


Figure 6.15 (b): TTE of Body, AFC/ KBFAFC, *Qvib,* Q_{gain} = 0.5

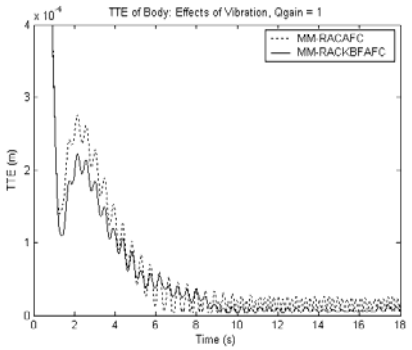


Figure 6.15 (c): TTE of Body, AFC/ KBFAFC, *Qvib,* Q_{gain} = 1

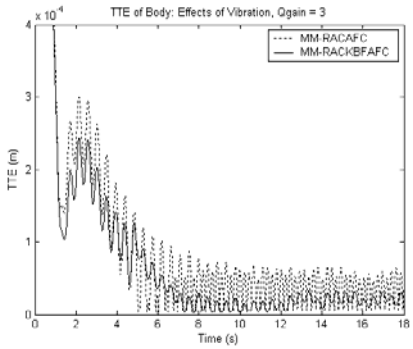


Figure 6.15 (d): TTE of Body, AFC/ KBFAFC, *Qvib,* Q_{gain} = 3

6.5 Conclusion

Three major conclusions can be drawn from the study carried out in this chapter:

- The procedure of knowledge investigation to knowledge processing performed in this study is proven very effective to simplify the fuzzy system design. The procedure is also helpful to design the proper fuzzy system applied as the inertia matrix estimator. In general, the knowledge that captured within the investigation, that is actual velocity-inertia matrix relation, is successfully applied as base to design the knowledge-based fuzzy for the intelligent active force control of the mobile manipulator.

- The design of partial KBF for each of subsystem, that are joints and wheels, is found very efficient to reduce the mathematical burden and the structure of fuzzy systems. Thus, the RACKBFAFC can be considered as an effective practical solution on robust motion control.

- From the simulation, the proposed MM-RACKBFAFC is successfully proven as a stable and robust motion control. Comparing to the RACAFC, the proposed scheme is much better in general robot operation including the presence of the disturbances in the form of pulses impact and vibration.

CHAPTER 7

EXPERIMENTAL MOBILE MANIPULATOR

7.1 Introduction

This chapter describes the practical development of the mobile manipulator. A complete experimental rig called MMAFCON (Mobile Manipulator AFC ON-line system) is designed and developed to demonstrate the feasibility and implementability of the investigated control methods proposed in the theories. The concept of mechatronics approach in designing the MMAFCON system is especially highlighted as it involves the full integration of the mechanical, electronics and computer control disciplines. For the initial stage of the practical implementation of the proposed schemes we have experimented the RAC and RACAFC. The experiment on the RAC only scheme is to show the effectiveness of the proposed simplified coordinates (x, y) and heading angle (φ) of nonholonomic mobile robot motion control using resolved acceleration control. Based on this then the RACAFC is experimented to demonstrate the powerful of the AFC scheme. Note that due to the technical constraints in the rig design such as imprecise mounting and improper specification of the mechanical, electrical and electronic components including motors and sensors, the precise tracking by the proposed (and improved) AFC scheme could not clearly be measured in practice. The experimentation of both the RACAFC and RACKBFAFC are done with such practical constraints.

7.2 Specifications of Mobile Manipulator

Generally, the MMAFCON system comprises both the mobile platform and two-link robot arms and is composed of the hardware and software components. The hardware is made up of mechanical, electrical and electronics elements including the personal computer based control system. The main interface between computer and the rig is accomplished using two units of the data acquisition card DAS1602. The software or the driver program is designed using a high level programming language, i.e., C language. For ease of use, executable files are built into the driver program and can be executed under the common personal computer disk operating system (DOS) environment.

The experimental rig developed in this study is constrained by a number of limitations as follows:

- Types of the motors are actually DC-servomotors with linear characteristics, however not exactly torque motors as needed.
- For the platform, the accelerometer cannot be mounted on the wheel as in the simulation due to impossibility to install this wired-type sensor at the angular side of the wheel.
- It is difficult to keep the weight of the robot, whether totally or on subsystems, as same as in the simulation, particularly due to the weight of the cable connection among the PC and robot that should not included to the total weight.
- The adjustment of center of gravity of the robot is also complicated. To solve it we adjusted the mounting position of the batteries.
- Frictions, backlash, and imprecise mounting of the rig are considered as unstructured dynamics and unknown disturbances.

Table 7.1 shows the specifications of the rig that was designed and developed in this study.

Table 7.1: Specifications of the rig

Parameters	Mobile Platform	Manipulator
Type	Differentially-Driven Mobile Robot (left & right)	Two-link Planar Arm
Mass of platform	48 kg	
Mass of wheel	1 kg (each)	
Half width of platform	0.4 m	
Radius of wheel	0.11 m	
Distance of point *G* to *F*	0.3 m	
Mass of link-1 (plus motor at joint 2)		3.45 kg
Length of link-1		0.4 m
Length of link-1 from joint-1 to centre of gravity		0.24 m
Mass of link-2 (plus gripper)		4.25 kg
Length of link-2		0.38 m
Length of link-2 from joint-2 to centre of gravity		0.21 m
Type of Motor joint-1		VEXTA: Motor AXHM450K Gear GFH4G100
Type of Motor joint-2		VEXTA: Motor AXHM230K Gear GFH2G100
Type of Motor wheel-L	VEXTA: Motor AXHM450K Gear GFH4G50	
Type of Motor wheel-R	VEXTA: Motor AXHM450K Gear GFH4G50	

The total weight of the mobile manipulator is around 55.70 kg including batteries with the weight of the manipulator is around 7.7 kg including the gripper. The length of the manipulator is 40 cm for link-1 and 38 cm for link-2. The distance between two wheels is 80 cm with the wheel radius of 11 cm.

The isometric view of the mobile manipulator rendered from the AUTOCAD® drawing is shown in Figure 7.1 (a), while the actual mobile manipulator and its current end-effector are shown in Figures 7.1 (b) and 7.1 (c) respectively.

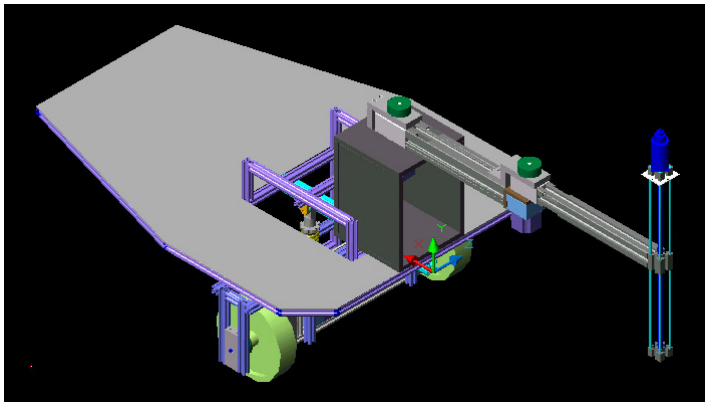


Figure 7.1 (a): Isometric view of the mobile manipulator

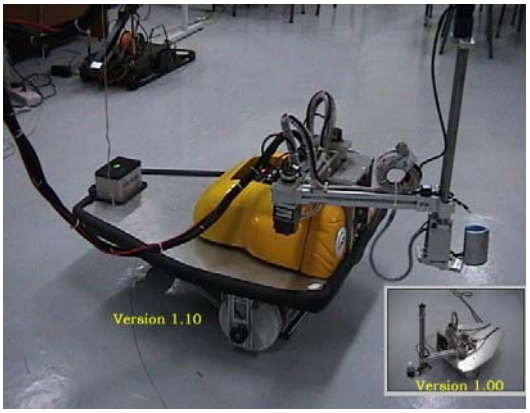


Figure 7.1 (b): A photograph of the mobile manipulator in action

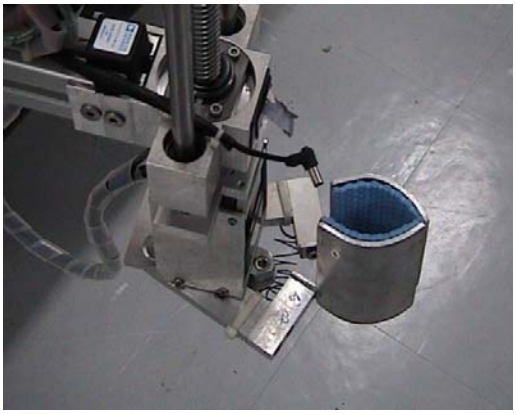


Figure 7.1 (c): Proposed gripper attached to the manipulator

In Figure 7.1 (c), the gripper that is attached to tip position is operated to grasp and move the object/load with maximum of 1 kg. The small window at the right corner in Figure 7.1 (b) shows the first development of the robot in which the grasper/gripper has been improved by redesigning the mechanical parts and replacing the motor with the more sophisticated ones (VEXTA Motor). Figure 7.1 (c) shows the current gripper design.

7.3 Personal Computer (PC) Based Controller

The controller of the mobile manipulator developed in this study is composed of two types, i.e. PC controller supported by DAS1602 interface cards, and

Embedded Controller System based on a Microchip Peripheral Interface Controller PIC16F877. These controllers can be selectively operated by setting the connectors attached at the robot's body. Figure 7.2 shows the outline of the experiment conducted for the PC based controller that is supported by two units of data acquisition cards DAS1602® manufactured by Keithley® Inc. The schematic diagram of the PC based controller of the mobile manipulator is shown in Figure 7.3.

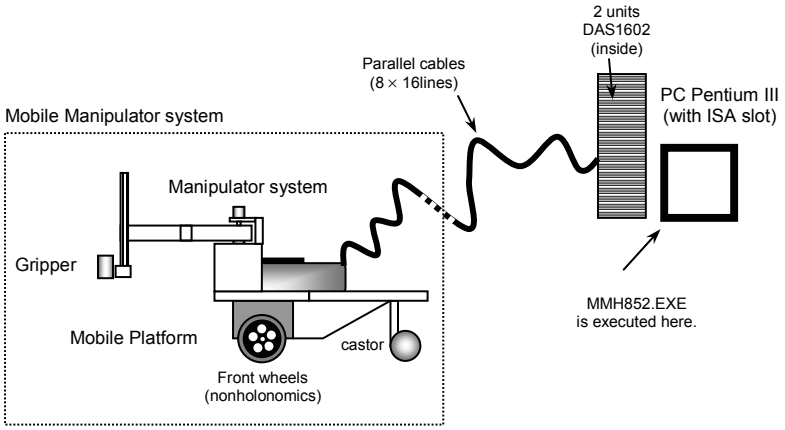


Figure 7.2: An experimental set-up for the PC-based controller

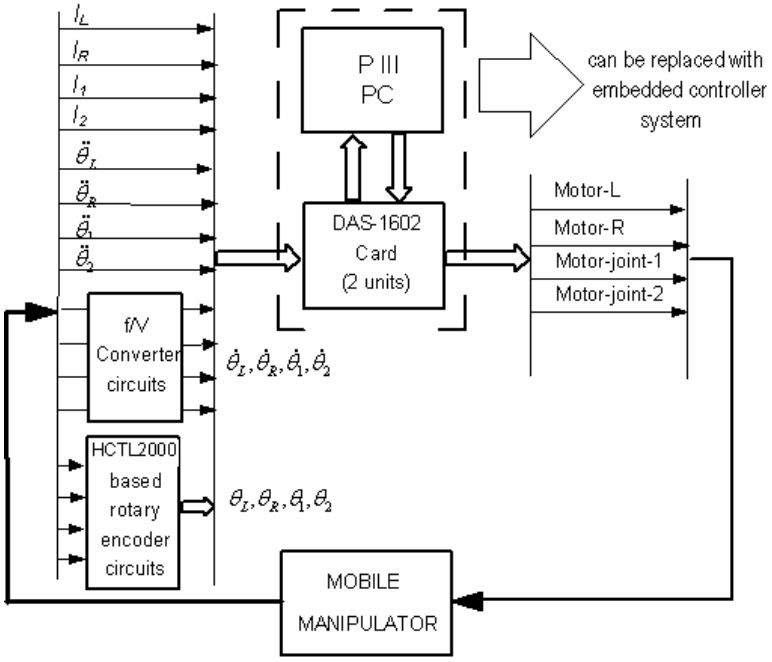


Figure 7.3: Schematic diagram of the PC based controller

Basically, the design of PC based controller must be able to organize all of the input and output functions of the mobile manipulator control as well as in the simulation. In total, the controller has four channels of parallel input functions for position variables, 12 channels of analog input functions for the velocity, acceleration and current sensors, and four channels of analog output functions for the actuators. For the position sensor, a frequency to voltage converter is applied to convert the pulses signals of the output of the rotary encoder to a linear analog signal for the purpose of measurement using the DAS1602 card. The motors used in the robot are DC servomotors that have embedded motor driver independently. The motor is driven by analog signal through the input of the driver which connects to the analog output of the DAS1602. Note that the motor moving direction must be driven by 1-bit digital output particularly, logic 1 for clock-wise, and logic 0 for counter-clock-wise direction.

7.3.1 Computer and Data Acquisition System (DAS) Card

The computer that is used as the controller is based on Pentium III PC – 1GHz with at least two units of EISA (*Extended Industrial System Association*) slots on the CPU board for the DAS1602 cards connection. Figure 7.4 shows the mounting of DAS1602 cards (two units) to the EISA slots.

It is noted that the power consumption must be sufficiently supplied by the PC power supply system. One card basically needs 1 to 3 Ampere when it runs. For the purpose of experiments, the analog input of ADC (*Analog to Digital Converter*) of the cards is set to bipolar function due to the outputs of sensors have analog voltage signals. While the analog output of DAC (*Digital to Analog Converter*) is set to unipolar function due to the motors must be commanded using unipolar voltage. The motors' moving direction is driven by the particular digital output that is one bit for each motor. The complete specification of the DAS1602 card can be found in the datasheet (Computer Boards, 1998).

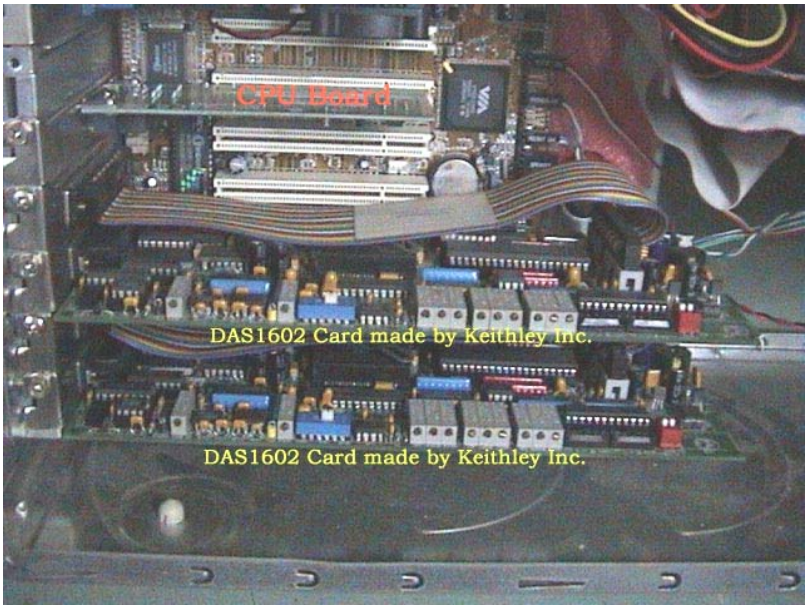


Figure 7.4: Two units of DAS1602 cards on the CPU Board

7.3.2 Frequency to Voltage Converter (f/V) Circuits

The motors used in the mobile manipulator are DC servomotor that has built in speed sensor with pulse signals output. Actually we can directly connect the speed sensor output to the digital input of the DAS1602 card. Consequently, the sensor output in pulses form must be converted to linear analog voltage in the program. However, it will be costly in computation and is time consumable.

To avoid the above problems, we implement a frequency to voltage converter as signal conditioning to the sensor so that the speed or velocity measurement can be performed through the analog input of the ADC of the DAS1602. Converting the sensor output to the linear analog signal basically can simplify the sensors reading processes in the program by only single instruction that is to read data from ADC. In contrast, pulses signal need to be converted using counters or similar function in programming or by functioning the programmable counter (hardware) of the interface. We use an IC LM2907 as converter for each of velocity signal of the motor. Figure 7.5 shows the (f/V) circuit.

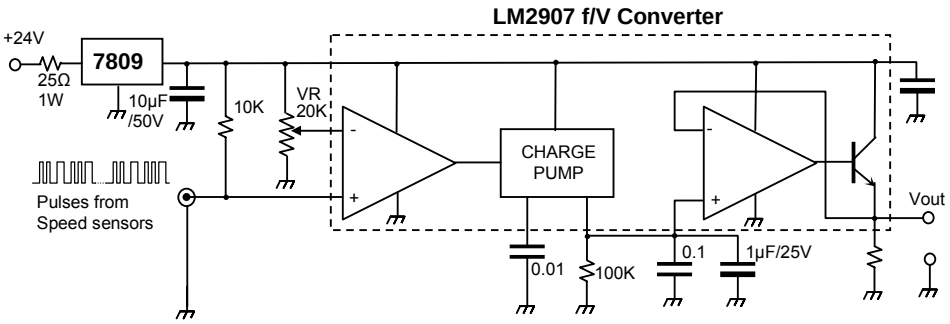


Figure 7.5: *Frequency to Voltage Converter* circuit

The IC LM2907 needs +9V supply system to work, while the robot system voltage is ±12V or ±24V. Therefore, an additional voltage regulator should be added to the circuit, as shown in Figure 7.5. The sensitivity of conversion can be adjusted through the variable resistor VR. It is highlighted that the sensor's wiring connections should use shielded (coaxial) cable to protect the signal from interferences or electromagnetic signals from the environments.

7.3.3 Rotary Encoder Circuits using HCTL2000

Figure 7.6 shows the complete circuit diagram of the signal conditioning system of the rotary encoder used in the robot controller. In the figure, the rotary encoders with function of CHA and CHB as the common signals are managed in which the four encoders can be read by the processor in real-time. It means the processor should use the effective way to read the position of each motor anytime when the sensor (position) reading is needed. Therefore the circuit is designed as particular section that can communicate to the processor using the specific bus connections. The bus consists of address bus (Port A of 82C55), data bus (Port B of 82C55), and handshaking signals (Port C of 82C55) as shown in the figure.

In Figure 7.6, CHA and CHB are Schmitt-trigger inputs which accept the outputs from a quadrature encoded source, such as the incremental optical shaft encoder. Two channels, A and B, nominally 90 degrees out of phase, are required.

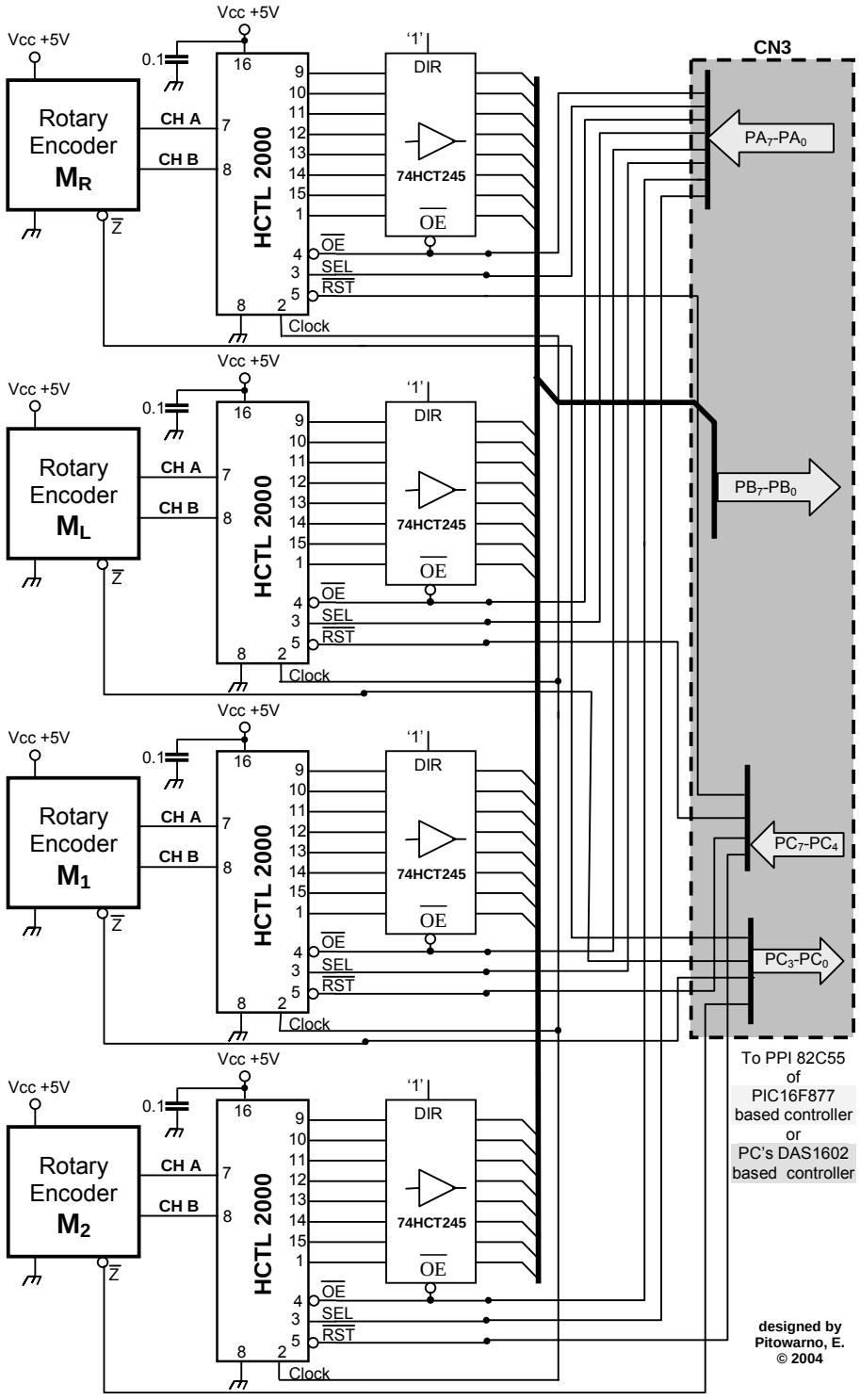


Figure 7.6: HCTL2000 based *Rotary Encoder Signal Conditioner*

The quadrature decoder samples the outputs of the CHA and CHB. Based on the past binary state of the two signals and the present state, it outputs a count signal

and a direction signal to the internal position counter. Then the data that were saved in the counter can be read by the processor anytime.

The processor needs to handshake to the encoder circuit when the reading is performed. Firstly, the processor sends SEL signal to the address of appropriate encoder. Then, $\overline{OE}$ is sent during the active SEL. From this point, the addressed encoder circuits will prepare a latched signal (latest data) at the data bus to ensure the data is valid. Now, the processor can safely fetch up the data from the bus.

7.3.4 Signal Conditioning Interfaces

Figures 7.7 and 7.8 show some views of the circuit boards containing the signal conditioners for all sensors and actuators. These boards were designed in such a way the wiring connection could easily be maintained for the purpose of debugging the program. Note that the connectors J-1 and J-2 in Figures 7.6 and 7.7 must be disconnected when the robot is controlled by the embedded controller system to ensure no signals interference among the PC and the embedded controller.

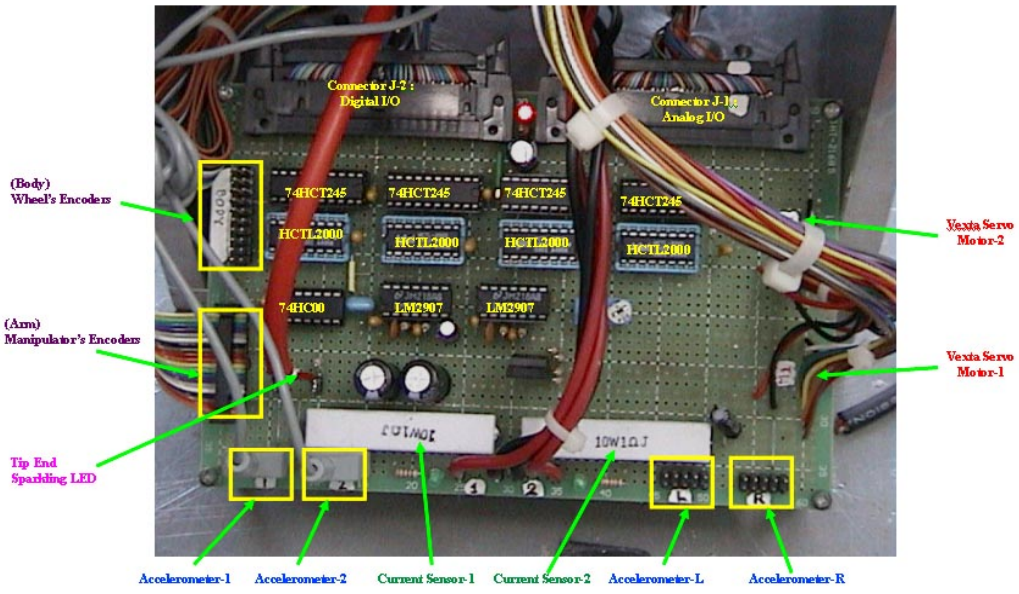


Figure 7.7: Signal conditioning interface for the mobile platform

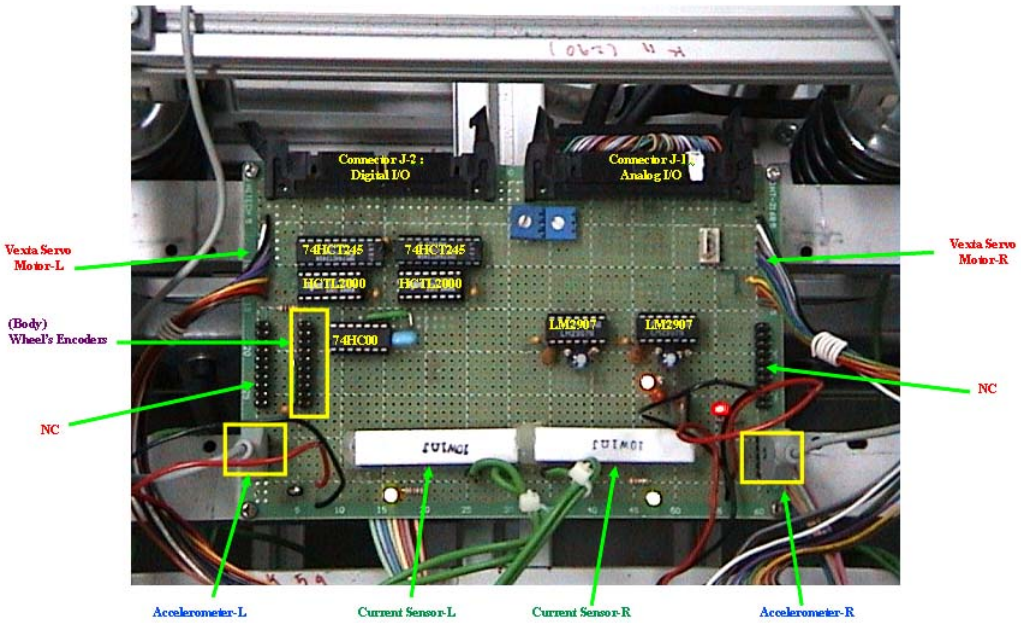


Figure 7.8: Signal conditioning interface for the manipulator

7.3.5 Programming Design

The program developed for the PC based controller is written in C. Essential program modules can be seen in **Appendix C**. To ensure a faster access time when the on-line feedback control program is running, the program is designed to run in the DOS (*Disk Operating System*) mode. It is known that all of the interrupt system programs in the Windows O/S based will be terminated (cancelled) when the PC is run under DOS. The PC-based program is developed with a real-time graphical display for convenience to the user. This feature is named RTM (*Real Time Monitor*). In the program development and testing at the initial stage, it is found that the quality of the power supply (voltage) to the system is very crucial. Note that the robot uses batteries to power the system. When the voltage or current of the batteries is low, the voltage reference used in the sensory-based measurement is affected, so that the sensor data is invalid. Consequently, the sensor must be recalibrated each time the power system becomes insufficient or weak. If not, the control parameters must be retuned to recover this adverse condition.

Therefore an additional procedure or function in a form of a computer program need to be designed to initially calibrate the power system before the main program is executed. The modified program is an executable file, MMH852.EXE in which it is able to readjust and recalibrate the sensory systems dynamically and automatically whenever the program is executed. Hence, the user can observe and evaluate the reference voltage of sensory system (shown as the ADC output value) that may exhibit undesirable change when the robot starts to run. The displayed value was then automatically used as reference value for computation in the program. From the experiments it is obtained that this procedure can substantially compensate the effects of the power system instability. Figure 7.9 shows the display on the computer screen when the calibration procedure is executed.

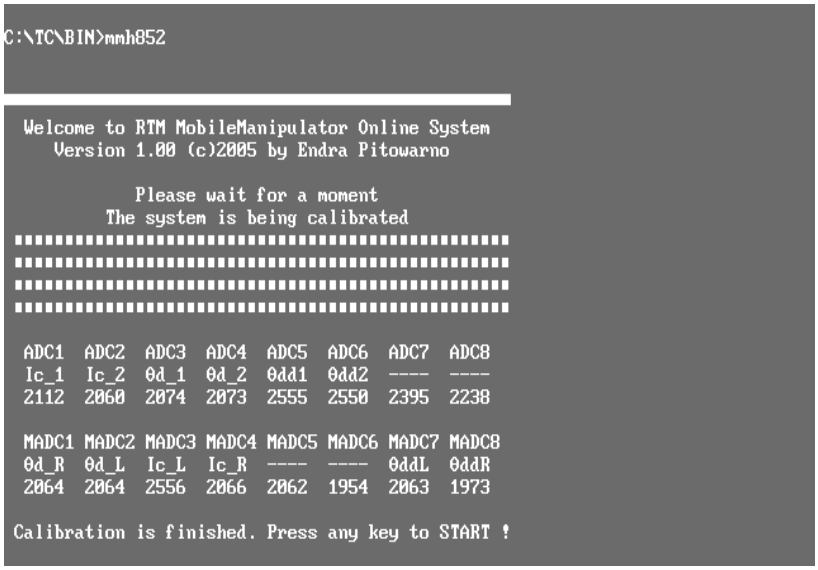


Figure 7.9: Display window of the calibration process through the program MMH852.EXE

For convenience in monitoring the robot operation and program debugging, the display is designed as a real-time graphical monitoring system. A sample of the display captured from the PC monitor is shown in Figure 7.10. In the figure, the left side is the real-time display of all sensor readings installed in the robot system. The middle portion of the graph shows the trace of the trajectory that is generated during the robot movement. The green line is the desired trajectory while the red one is the actual. The generated errors can also be monitored instantly but crudely (not scaled)

during the operation. The window at the bottom-right corner is the display for the selected control parameters, desired track and the initial conditions. In addition, the display is developed for three window settings related to the RAC, RACAFC, and RACKBFAFC schemes considered in the study.

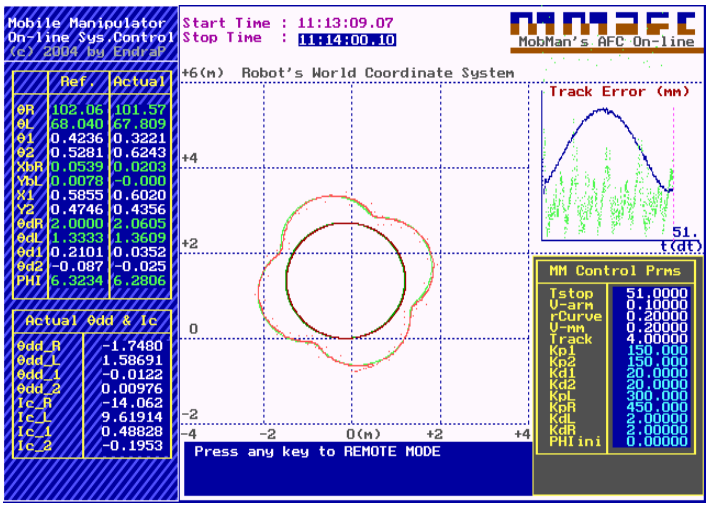


Figure 7.10: A sample of the display captured from the PC screen

7.4 Embedded Controller using PIC16F877

Figure 7.11 shows an outline of the experimental set-up for one of the earlier versions of the prototype of the complete autonomous mobile manipulator that is controlled by an embedded controller using PIC16F877 as the main CPU.

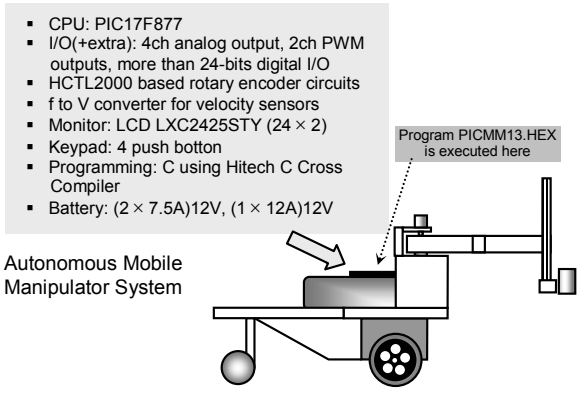


Figure 7.11: A descriptive diagram of the autonomous mobile manipulator

The microcontroller PIC 17F877 is selected as the embedded controller of the mobile manipulator in this study. The considerations are based on the following specifications (Microchip, 2001):

- high performance RISC CPU in fully static design at low price
- high speed CMOS FLASH/EEPROM technology
- low-power consumption at maximum 25 mA
- up to 8K x 14 words of FLASH Program Memory
- up to 368 x 8 bytes of Data Memory (RAM)
- up to 256 x 8 bytes of EEPROM Data Memory
- In-Circuit Serial Programming (ICSP) via two pins
- only 35 single word instructions to learn
- all single cycle instructions except for program branches which are two cycle
- wide operating voltage range: 2.0V to 5.5V
- many other advantages as described in the datasheet.

Figure 7.12 shows pins configuration of PIC16F877. This IC has five ports, namely, RA_5-RA_0, RB_7-RB_0, RC_7-RC_0, RD_7-RD_0, and RE_2-RE_0. Pins of RA_5, RA_3, RA_2, RA_1, RA_0, RE_2, RE_1, and RE_0 can be initialized as analog input port at 10-bit resolution. Thus, there are eight channels of analog input that can directly be connected to the sensors.

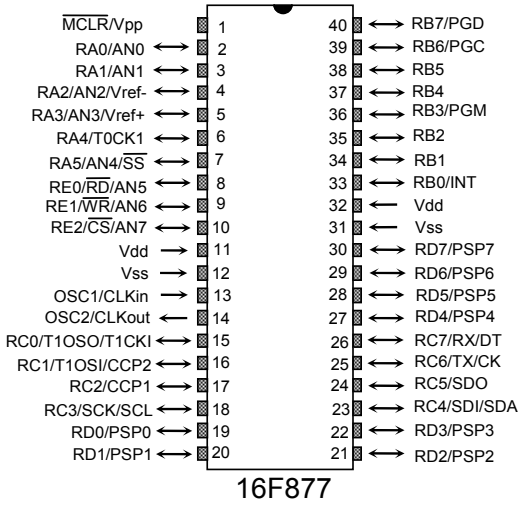


Figure 7.12: Pins configuration of PIC16F877[1]

[1] PIC16F877 is trademark of Microchip Inc.

The microcontroller can be programmed using a high level language (e.g. C) so that the equations derived from the theoretical investigation could be easily rewritten in the program. Compiling the program can be done using Hitech C Cross Compiler® from Hitech®[2] Software.

7.4.1 Embedded Controller Circuit

By considering the input/output configuration as shown in Figure 7.12, the embedded controller was then designed as shown in Figure 7.13. The sensors were directly connected to the chip through the connector CN1, except for the rotary encoders. In this case, we use an IC PPI (Programmable Peripheral Interface) 8255A as the interface of the encoder to the PIC. As can be seen in the figure, the PPI8255A is designed such that the PIC can communicate with the encoder using the specific bus connection through the connector CN2. Note that due to the limitation of the analog inputs of the PIC which has only eight channels, whereas the robot system needs 12 channels, then four channels (supposedly two channels for the current sensors and two channels for the accelerometers for the AFC function of the platform) are not connected. Consequently, the application of AFC is limited to the manipulator section only.

To drive the actuator system, an IC DAC0808 was used to perform the analog output and 1-bit digital output for commanding the moving direction. In this case, an IC 74HC377 was employed as the multiplexer to address the target motor. In total, there exists four channels of analog output to drive the actuator systems.

A mini display system was also designed using LCD (liquid crystal display) module, LXC2425STY that has two lines and 24 characters for each line. In this display, the status of the robot operation can be monitored. Figure 7.13 shows the complete circuit diagram of the embedded controller system for the autonomous mobile manipulator.

[2] Hitech C Cross Compiler is trademark of HI-TECH Software Pty Ltd., Australia.

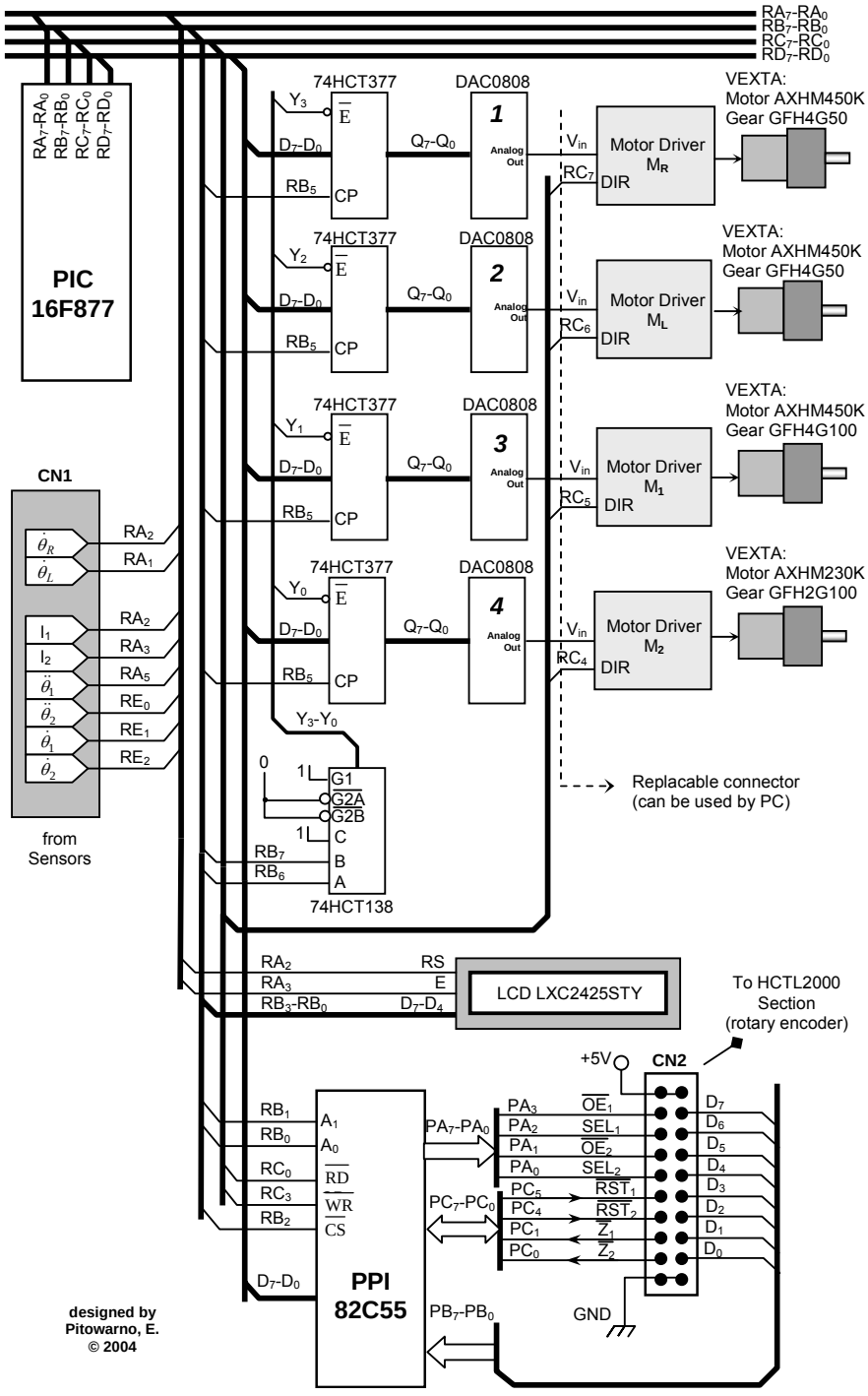


Figure 7.13: Circuit diagram of PIC16F877 based controller

7.4.2 Controller Board

Figures 7.14 (a) to (d) show the autonomous mobile manipulator, the mounting of the embedded controller, the PIC 16F877 based embedded controller and the power supply unit for the embedded controller.

Figure 7.14 (a): A view of the Autonomous Mobile Manipulator

Figure 7.14 (b): The mounting of the embedded controller

Figure 7.14 (c): The PIC 16F877 based embedded controller

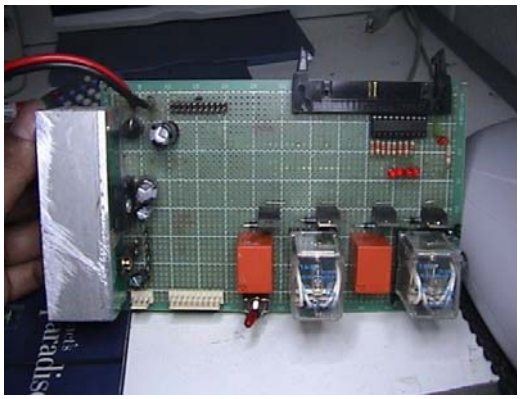


Figure 7.14 (d): Power Supply unit for the embedded controller

The controller, as seen in Figure 7.14 (c), is designed in one board system so that the wiring system could be minimized. The connectors attached on the board can be easily connected to the sensor and actuator systems using flat cables.

7.5 Experimental Results and Discussion

The experimental result highlighted in this discussion is based on the PC-based controller. Using PC-based controller, the measurement of the actual conditions of the robot operation is automatically performed by the program and the data is automatically recorded in the hard disk of the PC after program execution. In the embedded system, the data recording cannot be performed due to the limitation of the memory. As mentioned earlier, the R/W (read/write) memory capacity of the PIC is 2 Kilobytes only.

It is desired that the robot task is to move its platform in a circular motion with a curvature radius of 1.5 m, at a velocity of 0.2 m/s (at point F) and the initial heading angle orientation of 0 rad to the zero angle of the world *Cartesian* coordinate. The friction, backlash and imprecise mounting of the elements in the rig maybe considered as unstructured dynamics and unknown disturbances. Note that in this experiment, the AFC and KBFAFC are implemented to the manipulator only due to the use of wired-sensor type of the accelerometers.

Figures 7.15 and 7.16 show the results for the RACAFC and RACKBFAFC schemes respectively.

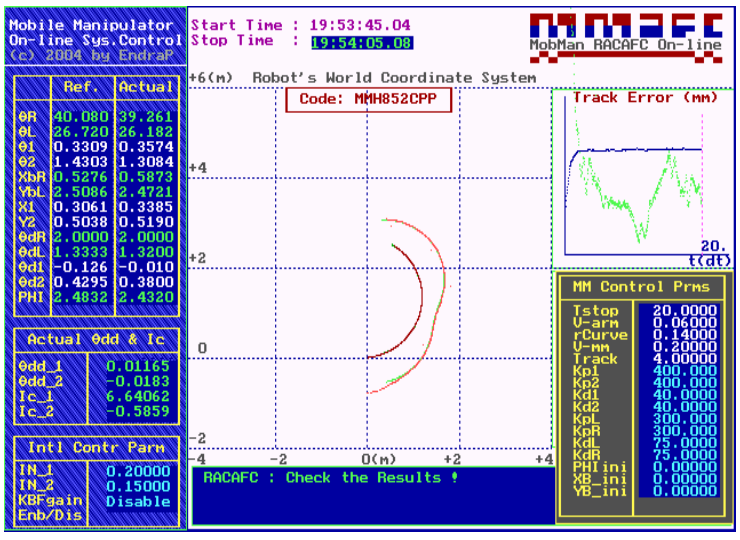


Figure 7.15: Experimental results of the RACAFC scheme

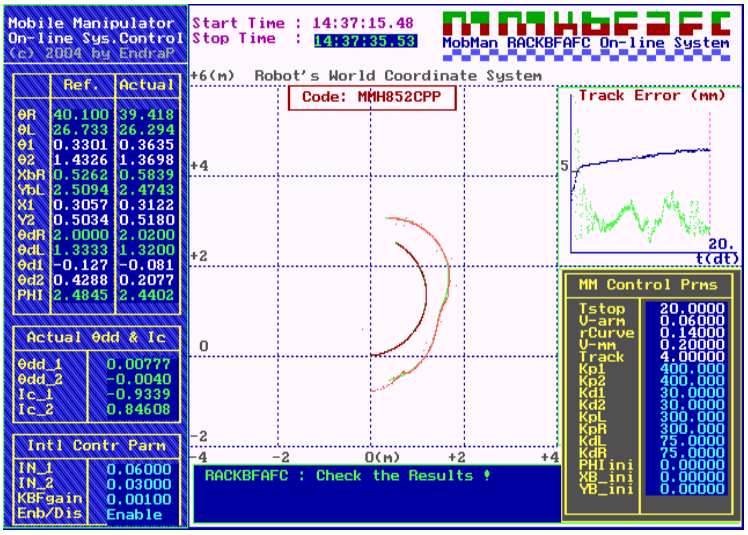


Figure 7.16: Experimental results of the RACKBFAFC scheme

From the instant visualization as shown in the window of *Track Error* in Figures 7.15 and 7.16, it can be deduced that the RACKBFAFC performs better than the RACAFC counterpart. The actual tracking errors at the tip position for both schemes can be seen in Figure 7.17 where the graphs are redrawn from the recorded data after program execution.

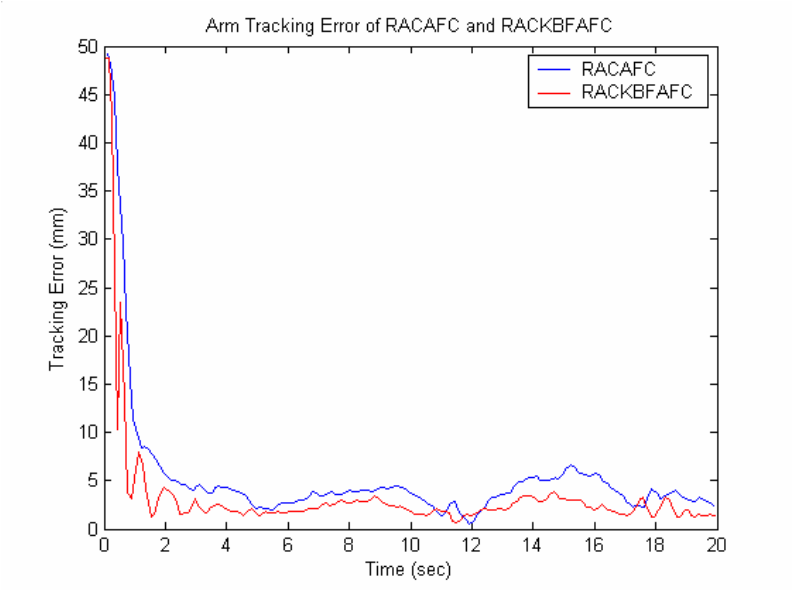


Figure 7.17: Tracking errors of the arm in the experiment of RACAFC and RACKBFAFC schemes

Figure 7.18 depicts the actual acceleration at the joints for the RACAFC and RACKBFAFC schemes, while the applied current of the motors can be seen in Figure 7.17. As can be seen, the actual acceleration in the experiment is very sensitive and tends to be spiky. It is probably due to the backlash of gearbox of the motors at the joints. It is highlighted that from the current signals, it can be deduced that there seems to be nonlinearity in the motors characteristic. In the range of around ±0.3A it seems the moving direction is abruptly changed. In fact, after the thresholds, the motors smoothly move and track the trajectory. It may be also due to the backlash of the gears and the fact that the motors driver circuit is nonlinear within the range of currents. It is reasonable due to the motor VEXTA used in the experiment is actually a brushless motor that needs offset current to start to move.

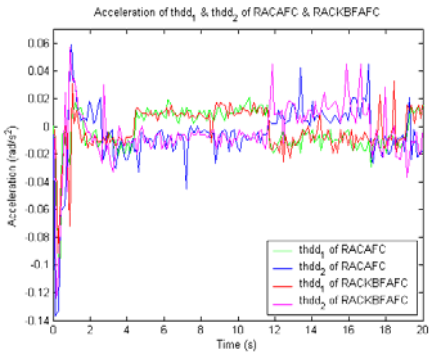


Figure 7.18: Actual acceleration at the joints, RACAFC and RACKBFAFC

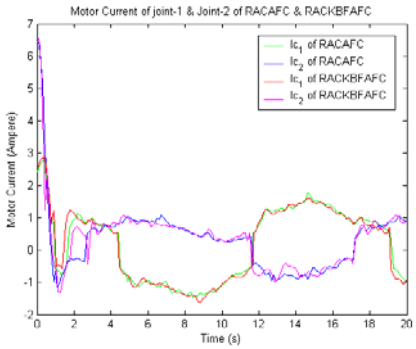


Figure 7.19: Actual motor currents at the joints, RACAFC and RACKBFAFC

7.6 Conclusion

From the experiments conducted, a number of conclusions can be drawn and are listed as follows:

- The autonomous mobile manipulator was successfully designed and developed in this study in the form of a working prototype but with very limited memory for programming.

- The experiments show the successful implementation of the AFC scheme in mobile manipulator motion control considering some practical constraints.

- In the experiment, the RACKBFAFC has been proven to perform better than the RACAFC. This is in line with the results obtained through the simulation study and hence validates the theoretical aspect of the research. However, further investigation has to be carried out by introducing other forms of disturbances and trajectories to enhance the system performance, particularly, in terms of robustness, stability and accuracy.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusion

It can be summarized that the specific objective of the project has been satisfactorily achieved in that an ultimate working prototype of an intelligent mobile transporter with AFC capability operating in an appropriate ‘factory’ setting was successfully designed and developed. Both theoretical and practical aspects of the research were rigorously and systematically executed.

The simulation study on the mobile robot system was performed considering various AFC-based schemes with a number of intelligent elements and disturbances incorporated into the schemes - all pointing towards the same direction (conclusion), i.e. all producing the expected results which indeed verify the robustness and effectiveness of the AFC algorithm. The RAC knowledge-based fuzzy AFC (RACKBFAFC) scheme was particularly very efficient and accurate in performing the trajectory tracking tasks under various operating and loading conditions. A new method of tuning the estimated inertia matrix of the mobile robot system through KBF technique was realized.

The practical system was successfully developed in the form of a working prototype mobile manipulator employing a number of control schemes that are theoretically defined in the earlier chapters. Only RAC, RACAFC and RACKBFAFC schemes were realized and implemented into the prototype by way of

rigorous programming, interfacing and control. The results obtained from the experiments do complement and ‘tally’ with those of the theoretical (simulation) counterparts implying that the schemes were successfully validated even though it was found that the experimental findings were rather crude and much lower in accuracy, obviously due to a number of physical constraints. Indeed, a complete mechatronic approach and design was fully utilised and implemented here.

One of the innovative features that is incorporated into the research is the application of virtual reality (VR) technique to the mobile manipulator that enables user to visualise the operation of the system in different perspective. The simulator is tested and evaluated considering a factory layout in the form of computer integrated manufacturing (CIM) environment. A collision avoidance algorithm was successfully implemented into the simulator making it more autonomous and intelligent.

8.2 Recommendations for Future Works

A number of recommendations for future works is listed as follows:

- Perform a rigorous stability analysis of the mobile manipulator with various AFC-based schemes.
- Simulate system using other forms of operating and loading conditions to further investigate the system robustness and accuracy in performing its tasks. This may include different trajectories (paths), terrains, velocities, payloads, tasks and other forms of external disturbances.
- Add more intelligence to the system be it through software program (incorporating other articicial intelligence (AI) techniques) or hardware by way of using intelligent (or smart) sensors and electronics.
- Solve some of the current problems related to physical hardware constraints such as appropriate mounting of sensors and actuators and also using more accurate sensors.

- Further practical experiments need to be carried out to obtain other useful characteristics or behaviours of the mobile manipulator.
- A number of end-effectors (grippers or tools) can be designed, developed and installed to the manipulator. This may be done in total isolation of the system at the initial stage and may involve the use of different controller (e.g. PLC, microcontroller or even PC-based control). However, synchronisation need to be done to ensure that the overall system runs efficiently.
- The mobile manipulator can be further tested in actual manufacturing environment to perform specific tasks.
- A self charging gadget for the battery can be designed and added to the system to enhance its innovative feature.
- The VR application can be improved to take into account more immersive features and different work settings/environments.

REFERENCES

Antonelli, G., Chiaverini, S., Fusco, G. (2002). Experiment of Fuzzy Real Time Path Planning for Unicycle-Like Mobile Robots Under Kinematic Constraints. *Proceeding of the 2002 IEEE International Conference on Robotics and Automation.* Washington: IEEE, 2147-2152.

Aycard, O., Charpillet, F. and Haton, J. -P. (1997). A New Approach to design Fuzzy Controllers for Mobile Robots Navigations. *Proceeding of the 1997 IEEE International Conference on Computational Intelligence in Robotics and Automation*. 68-73.

Aydin, S. and Temeltas, H. (2002). A Novel Approach to Smooth Trajectory Planning of a Mobile Robot. *Proceedings of 7th International Workshop on Advanced Motion Control*, pp. 472-477, Maribor, Slovenia, July 3-5, 2002.

Bayle, B., Fourqet, J. Y. and Renaud, M. (2001). Manipulability Analysis for Mobile Manipulators. *Proc. IEEE Int'l Conf. on Robotics & Automation*. 1251-1256.

Campa, R., Kelly, R., and Garcia, E. (2001). On Stability of the Resolved Acceleration Control. *Proc. IEEE International Conf. on Robotics & Automation*. 3523-3528.

Ciliz, M. K. (2005). Rule base Reduction for Knowledge0based Fuzzy Controllers with Application to a Vacuum Cleaner."

Colbaugh, R. (1998). Adaptive Stabilization of Mobile Manipulator. Proc. American Control Conf. 1-5.

Cook, C. C. and Ho, C. Y. (1985). The Application of Spline Functions to Trajectory Generation for Computer-Controlled Manipulators. In: Aleksander, I. *Computing Techniques for Robots*. London: Kogan Page Ltd., 102-110.

D'souza, A., Vijayakumar, S., and Schaal, S. (2001). Learning inverse Kinematics. *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems*. 298-303.

Engineering Animation, Inc. (1999). *WorldToolKit® Reference Manual Release 9*, Mill Valley (USA).

Fierro, R. and Lewis, F. L. (1995). Control of Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics. *Proc. 34th Conf. On Decision and Control*. 3805-3810.

Fierro, R. and Lewis, F. L. (1998). Control of Nonholonomic Mobile Robot Using Neural Networks. *IEEE Trans. Neural Networks*. **6**(4). 589-600.

Foley, J. D., Dam, A. V., Feiner, S. K. and Hughes, J. F. *Computer Graphics: Principles and Practice – Second Edition in C*. 2nd edition. United States of America: Addison-Wesley Publishing Company, Inc. 1996.

Franklin, G., Powell, F., David, J. and Naeini, A. E. (2002). *Feedback Control of Dynamic Systems* – 4rd Ed. New Jersey: Prentice – Hall, Inc.

Fu, K.S., Gonzales, R.C. and Lee, C.S.G. (1987). *Robotics: Control, Sensing, Vision, and Intelligence*. New York: McGraw-Hill, Inc.

Fukao, T., Nakagawa, H., and Adachi, N. (2000). Adaptive Tracking Control of a Nonholonomic Mobile Robot. *IEEE Trans. Robotic and Automation*. **16**(5). 609-615.

Godler, I., Honda, H., and Ohnishi, K. (2002). Design Guidelines for Disturbance Observer's Filter in Discrete Time. *Proc. International Workshop on Advanced Motion Control*. **1**. 390-395.

Godler, I., Inoue, M., Ninomiya, T., and Yamashita, T. (1999). Robustness Comparison of Control Schemes with Disturbance Observer and with Acceleration Control Loop. *Proc. IEEE ISIE*. 1035-1040.

Growe, S., Schröeder, T., and Liedtke, C.Ë. (2000). Use of Bayesian Networks as Judgement Calculus in a Knowledge-Based Image Interpretation System. *IAPRS Journal*. **33**.101-110.

Hassanzadeh, I., Khanmohammadi, S., Jiang, J., and Alizadeh, G. (2002). Implementation of a Functional Link-net ANFIS Controller for a Robot Manipulator. *Proc. Int'l Workshop on Robotic Motion and Control*. 399-404.

Hewit, J.R. and Burdess (1981). Fast Dynamic Decoupled Control for Robotics Using Active Force Control. *Trans. Mechanism and Machine Theory*. **16**(5). 535-542.

Hewit, J. R. and Marouf, K. B. (1996). Practical Control Enhancement via Mechatronics Design. *IEEE Trans. Industrial Electronics*. **43**(1). 16-22.

Hewit, J.R. and Morris, J.R. (1999). Disturbance Observation Control with Estimation of The Inertia Matrix. *Proc. IEEE/ASME, International Conference on Advanced Intelligent Mechatronics*. 753-757.

Hildebrand, L. and Fathi, M. (2004). Knowledge-Based Fuzzy Color Processing. *IEEE Trans. Systems, Man, and Cybernetics – Part C: Applications and Reviews*. **34**(4). 499-505.

Hu, Y., Yang, S. X. (2003). A Fuzzy Neural Dynamic Based Tracking Controller for a Nonholonomic Mobile Robot. *Proceedings of the 2003 IEEE/ASME International Conference on Advance Intelligent Mechatronics (AIM 2003)*. 205-210.

Ignizio, J. P. (1991). *Introduction To Expert Systems: The Development and Implementation of Rule-Based Expert System*. Singapore: McGraw-Hill, Inc.

Jang, J. S. R. (1993). ANFIS: Adaptive Network-Based Fuzzy Inference System. *IEEE Trans. Systems, Man, and Cybernetics*. **23**(3). 665-685.

Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T. (1990). A Stable Tracking Control Method for an Autonomous Mobile Robot. *Proc. IEEE Int. Conf. Robot and Automation*. 384-389.

Kim, S. W. and Lee, J. J. (1993). Resolved Motion Rate Control of Redundant Robots using Fuzzy Logic. *Proc. 2nd IEEE Int'l Conf. Fuzzy Systems*. **1**. 333-338.

Kircanski, M. and Kircanski, N. (1998). Resolved rate and acceleration control in the presence of actuator constraints. *IEEE Control Systems Magazine*. 18(1). 42-47.

Kolmanovsky, I. and McClamroch, N. H. (1995). Development in Nonholonomic Control Problems. *IEEE Control Systems*. 20-36.

Komada, S., Kimura, T., Ishida, M., and Hori, T. (1996). Robust Position Control of Manipulator based on Disturbance Observer and Inertia Identifier in Task Space. *Proc. International Workshop on Advanced Motion Control*. **1**. 225-230.

Komada, S. and Ohnishi, K. (1990). Force Feedback Control of Robot Manipulator by the Acceleration Tracing Orientation Method. *IEEE Trans. Industrial Electronics*. **37**(1). 6-12.

Kuwata, Y. and Yatsu, M. (1997). Managing Knowledge using a Semantic Network. *Proc. AAAI Spring Symposium: Artificial Intelligence in Knowledge Management,* Stanford University. 94-98.

Lewis, F. L., Abdallah, C. T., Dawson, D. M. (1993). *Control of Robot Manipulators*. New York : Macmillan Publishing Company.

Liao, S. H. (2005). Expert System Methodologies and Applications – a Decade Review from 1995 to 2004. *Journal of Expert System with Applications*. 28. 93-103.

Lin, S. (2001). *Robust and Intelligent Control of Mobile Manipulators.* University of Toronto: PhD Thesis.

Lin, S. and Goldenberg, A. A. (2001). Neural-Network Control of Mobile Manipulator. *IEEE Trans. Neural Networks*. **12**(5). 1121-1133.

Luh, J. Y. S., Walker, M. W., and Paul, R. P. C. (1980). Resolved-Acceleration Control of Mechanical Manipulator. *IEEE Trans. Automatic Control*. **25**. 486-474.

Mar, J. and Lin, F. J. (2001). An ANFIS Controller for the Car-Following Collision Prevention System. *IEEE Trans. Vehicular Technology*. **50**(4). 1106-1113.

Marakas, G. M. (1999). *Decision Support Systems in The Twenty-First Century*. New Jersey: Prentice Hall.

Mitchell, T. (1997), *Machine Learning*. New York: McGraw Hill.

Mohri, A., Furuno, S., Iwamura, M. and Yamamoto, M. (2001). Sub-Optimal Trajectory Planning of Mobile Manipulator. *Proc. IEEE Int'l Conf. on Robotics & Automation*. 1271-1276.

Muir, P.F. and Neuman, C.P. (1990). Resolved Motion Rate and Resolved Acceleration Servocontrol of Wheeled Mobile Robots. *Proc. IEEE Int'l Conf. Robotics and Automation*. **2**. 1133-1140.

Musa Mailah (1998). *Intelligent Active Force Control of a Rigid Robot Arm Using Neural Network and Iterative Learning Algorithms*. University of Dundee, UK: Ph.D Thesis.

Musa Mailah and Hooi N. B. (2000). Intelligent Active Force Control of a Three-Link Manipulator Using Iterative Learning Technique. *Jurnal Teknologi*, UTM. 46-69.

Musa Mailah and Nurul Izzah Ab Rahim. (2000). Intelligent Active Force Control of a Robot Arm Using Fuzzy Logic. *Proc. IEEE International Conference on Intelligent Systems and Technologies TENCON 2000*, Kuala Lumpur. **2**. 291-297.

Nanayakkara, N. D. and Samarabandu, J. (2003). Unsupervised Model Based Image Segmentation Using Domain Knowledge-Based Fuzzy Logic and Edge Enhancement. *Proc. ICME*. 577-580.

Negnevitsky, M. (2002). *Artificial Intelligence – A Guide to Intelligent Systems*. London: Addison-Wesley.

Ohnishi, K. (1995). Industry Applications of Disturbance Observer, *Procs. of Int'l Conference on Recent Advances in Mechatronics*, Istanbul, Turkey, 14-16.

Olivares, M., Albertos, P., and Sala, A. (2001). Open-loop Fuzzy Control: Iterative Learning. *Proc. IFAC Workshop: Advanced Fuzzy/Neural Control AFNC'01*. 87-92.

O'neil, K. A., Chen, Y. C., and Seng, J. (1997). Removing Singularities of Resolved Motion Rate Control of Mechanism, including Self-Motion. *IEEE Trans. Robotics and Automation*. 13(5). 741-751.

Ouchi, Y. and Tazaki, E. (1998). Heuristic Approach to Topology Generation for Knowledge based Fuzzy Petri Nets. *Proc. 2nd Int'l Conf. Knowledge-Based Intelligent Electronics System.* 331-334.

Papadopoulos, E. and Poulakakis, I. (2001). Planning and Obstacle Avoidance for Mobile Robots. *Proc. IEEE Int'l Conf. Robotics and Automation*. 3967-3972.

Perrier, C., Dauchez, P. and Pierrot, F. (1998). A Global Approach for Motion Generation of Non-Holonomic Mobile Manipulator. *Proc. IEEE Int'l Conf. on Robotic & Automation.* 2971-2976.

Piazzi, A. and Guarino, L.B.C. (2000), Quintic G^2-splines for Trajectory Planning of Autonomous Vehicles, *IEEE Proc. Of Intelligent Vehicles Symposium,* 198-203, Dearborn.

Pitowarno, E., Musa Mailah and Hishamuddin Jamaluddin. (2001). Trajectory Error Pattern Refinement of A Robot Control Scheme Using A Knowledge-Based Method. *Proc. (in a CDROM) International Conference on Information, Communications & Signal Processing (ICICS 2001).* Singapore. P0301.

Pitowarno, E., Mailah, M., and Jamaluddin, H. (2002). Knowledge-Based Trajectory Error Pattern Method Applied to an Active Force Control Scheme. *IIUM Engineering Journal*. **3**(1). 1-15.

Pourboghrat, F. (2002). Exponential Stabilization of Nonholonomic Mobile Robot. *Computer and Electrical Engineering.* **28**. 349-359.

Rhee, F. V. D., Lemke, H. R. V. N., and Duckman, J. G. (1990). Knowledge Based Fuzzy Control of System. *IEEE Trans. Automatic Control*. **35**(2). 148-155.

Schmitt, G.N. (1993), Virtual Reality in Architecture in *Virtual Worlds and Multimedia,* ed. Thalmann, N.M. and Thalmann, D., John Wiley & Sons Ltd., England.

Sharir, M. (1989), Algorithmic Motion Planning in Robotics, *IEEE*_____, p.9-20.

Soucek, B. (1989). *Neural and Concurrent Real-Time Systems: The Sixth Generation*. New York: John Wiley & Sons, Inc.

Stone, R.J. (1995), The Reality of Virtual Reality, *World Class Design to Manufacture*, Vol. 2, No. 4, p.11-17.

Sugar, T. G. and Kumar, V. (2002). Control of Cooperating Mobile Manipulators. *IEEE. Trans. Robotic and Automation*. 18(1). 94-103.

Tanner, H. G. (2003). Nonholonomic Navigation and Control of Cooperating Mobile Manipulators. *IEEE Trans. Robotic and Automation*. 19(1). 53-61.

Uchiyama, M. (1989). Control of Robot Arms. *Trans. Japan Society of Mechanical Engineers*. III. **32**(1). 1-9.

Wang, C. C. and Kumar, V. (1993). Velocity Control of Mobile Manipulators. *Proc. IEEE Int'l Conf. on Robotics and Automation*. **2**. 713-718.

Yamamoto, Y. and Yun, X. (1996). Effect of the Dynamics Interaction on Coordinated Control of Mobile Manipulators. *IEEE Trans. Robotic and Automation.* **12**(5). 816-824.

Yasuda, G. and Takai, H. (2001). Sensor-Based Path Planning and Intelligent Steering Control of Nonholonomic Mobile Robot. *Proc. 27th Annual Conf. IEEE Industrial Electronics Society, IECON*. 317-322.

Young-Tack, P., and Wilkins, D. C. (1992). Representation and Control of Knowledge-Bases for Support of Multiple Task. *Proc. IEEE Conf. in Artificial Intelligence for Applications*. Monterey, CA.

Zadeh, L. A. (1988). Fuzzy Logic. *IEEE Computer*. 83-93.

Zheng, J.M., Chan, K.W. and Gibson, I. (1998), Virtual Reality, *IEEE Potentials*, Vol. 17, No. 2, p.20-33.

Zulli, R., Fierro, R., Conte, G. and Lewis, F.L. (1995). Motion Planning and Control for Non-Holonomic Mobile Robots, *Proceedings of IEEE International Symposium on Intelligent Control*, pp. 551-557, Monterey, CA, USA, August 27-29, 1995.

Appendix A

Notes on Heuristic Search Algorithms

A.1 Graph Searching Algorithm

Through a priori generated search space, a graph searching algorithm is required for the searching of a sequence of collision-free actions that will lead the mobile robot to the destination point with minimum cost. A number of graph searching algorithms can be applied for the findings of the collision-free nodes, namely breadth first algorithm, depth first algorithm, Dijkstra's method, Best-First-Search (BFS) algorithm, and A* heuristic algorithm. In this research, comparison has been conducted between Dijktra's method, Best-First-Search (BFS) algorithm, and A* heuristic algorithm. Generally, Dijkstra's method works in progressive stages to calculate the total cost of the shortest path from a single vertex to all other vertices [71]. The algorithm for Dijkstra's method is as shown in Figure 3.8. The graph searching begins with the initialization of the search. The total cost, F for initial vertex and all other vertices are set to 0 and infinity respectively. After the initialization of the search, the current vertex, V is marked as **visited**. The total cost for all the vertices adjacent to V, F_{child} is then adjusted through the summation of the total cost of V, F_V with the cost of the edge to the adjacent vertex, $F_{V\text{-}child}$. If the total cost for the adjacent vertex is less than the previously stored total cost, then the total cost for the child vertex is adjusted, or else it is left unchanged. All the vertices are then determined for their status. Vertex with the least total cost and **unvisited** will be selected as V, and the process is iterated until all the vertices are visited.

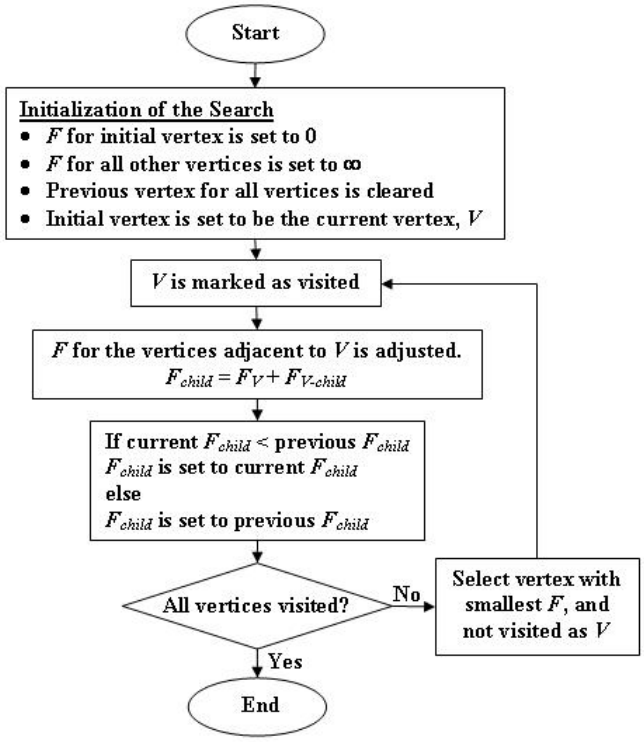


Figure A.1: Flow chart of Dijkstra's method

BFS algorithm works in a similar way as Dijkstra's method except that it practices certain heuristic features in the selection of successors. Instead of selecting the vertex closest to the starting point as shown in Dijkstra's method, BFS algorithm tends to choose heuristically the vertices which are closest to the goal as the successors. In brief, BFS is not guaranteed to find a shortest path as Dijkstra's method does, but BFS requires less computation time in solving the graph searching because fewer vertices in the graph are explored (LaValle, 2004). However, BFS is 'greedy' in the sense that it will always moves towards the goal although the selected successors are not on the right path. Therefore, in order to compromise the best features of these two algorithms, i.e. Dijkstra's method and BFS algorithm, A* heuristic algorithm has been introduced to search for the most optimum path. In general, heuristics are criteria, methods or principles which are adopted for the decision making in order to generate the most effective goal achieving approach (Pearl, 1984). The main purpose for the application of heuristics are to compromise between two requirements: to make the decision making rules simpler and at the same time discriminate correctly between good and bad choices. Usually, AI solvers employ heuristics in two basic situations (Luger, 2002), where

(i) the problem may not have an exact solution due to inherent ambiguities in the problem statement or available data.
(ii) the problem may have an exact solution, but the computational cost of finding for the solution may be prohibitive in terms of excessive computational times or limited computational resources.

In this research, A* heuristic algorithm, which has actually combined the best features of depth-first and Dijkstra's algorithms has been applied for the searching of the collision-free path. The A* heuristic algorithm is shown as follows:

1. Put the start node N_s on Open List
2. If Open List is empty, exit with failure.
3. Remove from Open List and place on Closed List a node N_i for which the cost function, $f(N_i)$ is minimum.
4. If N_i is a goal node N_G, exit successfully with the solution obtained by tracing back the pointers from N_i to N_s.
5. Otherwise expand N_i, generating all its successors, and attach to them pointers back to N_{i-1}. For every successor N_{i+1}:
5.1 If N_{i+1} is not on Open List or Closed List, the heuristic function, $h(N_{i+1})$ is estimated before the total cost of the action from N_i to N_{i+1} is obtained.
5.2 If N_{i+1} is already on Open List or Closed List, its pointer is directed along the path which yields the lowest $g(N_{i+1})$.
5.3 If N_{i+1} requires pointer adjustment and it is found on Closed List, then it is reopened.
6. Go to step 2

The flow chart of the A* heuristic search algorithm is illustrated in Figure A.1. From step 5.1, the total cost, $f(N_{i+1})$ is calculated as follows:

$$f(N_i+1) = g(N_{i+1})+h(N_{i+1}) \text{ , and} \quad \text{(A.1)}$$

$$g(N_i+1) = g(N_i) + c(N_i,N_{i+1}) \quad \text{(A.2)}$$

In equation (A.2), $c(N_i,N_{i+1})$ is the cost of action moving from N_i to N_{i+1}. In this research, the heuristic function $h(N_{i+1})$ is estimated from the *Euclidean* distance between the successor, N_{i+1} and the goal, N_G. The estimation of $h(N_{i+1})$ is expressed as follows:

$$h(N_{i+1}) = \sqrt{(x_{Ni+1} - x_{NG})^2 + (y_{Ni+1} - y_{NG})^2} \quad \text{(A.3)}$$

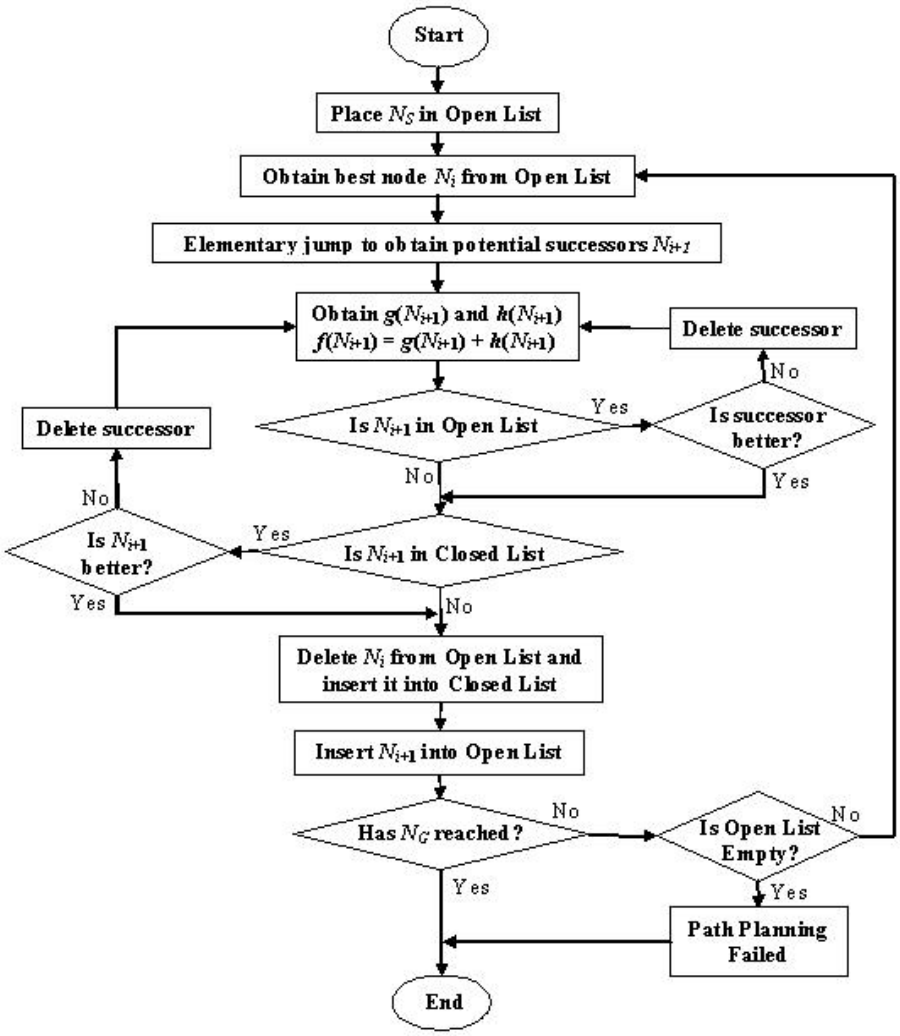


Figure A.3: Flow chart of the A* heuristic search algorithm

Appendix B

Computer Program for the Development of Virtual Environment (VE)

```
#include "matlab.h"
#include "simulator.h"
#include "wt.h"
#include "Model.hpp"
#include "Astar.hpp"
#include "Solution.hpp"

#define MapHeight 164
#define MapWidth 200

Solution *Final=new Solution;

//Main Program
int main()
{
        //Unvierse setup
        WTuniverse_new(WTDISPLAY_DEFAULT,WTWINDOW_DEFAULT);
        WTuniverse_seteventorder(4,event_array);
        WTuniverse_setbgrgb(20,10,60);
        root=WTuniverse_getrootnodes();
        WTlightnode_load(root,"lights");

        //Building up the scene
        MainWin=WTuniverse_getwindows();
        WTwindow_setposition(MainWin,0,0,1280,974);
        camera1=WTuniverse_getviewpoints();
        mouse=WTmouse_new();
        mouselink=WTmotionlink_new(mouse,camera1,WTSOURCE_SENSOR,
           WTTARGET_VIEWPOINT);
        TopView=WTviewpoint_new();

        //Setting viewpoint for each window
        WTwindow_setviewpoint(MainWin,camera1);

        //Repositioning the viewpoint for main window
        pos[0]=-750.0f; pos[1]=-100.0f; pos[2]=300.0f;
        WTviewpoint_setposition(camera1,pos);
        dir[0]=0.0f; dir[1]=0.0f; dir[2]=1.0f;
        WTviewpoint_setdirection(camera1,dir);

        //Repositioning the viewpoint for Orthographic view
        posTop[0]=-748.0f; posTop[1]=-904.0f; posTop[2]=617.0f;
        WTviewpoint_setposition(TopView,posTop);
        dir[0]=0.0f; dir[1]=1.0f; dir[2]=0.0f;
        WTviewpoint_setdirection(TopView,dir);

        //Adjusting the sensitivity of mouse
        WTsensor_setsensitivity(mouse,(float)0.01*WTnode_getradius(root));

        //Setting Up for the Virtual Environment
        SceneGraph();

        WTuniverse_setrendering(WTRENDER_BEST);
        WTuniverse_setactions(Actions);
```

```
	WTkeyboard_open();
	WTuniverse_ready();
	WTuniverse_go();
	WTuniverse_delete();
	return 0;
};
//Function
void SceneGraph(void)
{
	//Grouping all facilities into a group node
	facility=WTgroupnode_new(root);

	SepCup1=WTsepnode_new(facility);
	cupboard1=WTmovnode_load(SepCup1,"cupboard1.3ds",1.0f);
	pos[0]=0.0f; pos[1]=0.0f; pos[2]=182.0f+160.0f;
	WTnode_settranslation(cupboard1,pos);
	WTnode_axisrotation(cupboard1,Y,(float)3.142,WTFRAME_LOCAL);

	SepCup2=WTsepnode_new(facility);
	cupboard2=WTmovnode_load(SepCup2,"cupboard2.3ds",1.0f);
	pos[0]=0.0f; pos[1]=0.0f; pos[2]=90.0f+377.0f;
	WTnode_settranslation(cupboard2,pos);
	WTnode_axisrotation(cupboard2,Y,(float)3.142,WTFRAME_LOCAL);

	SepCup3=WTsepnode_new(facility);
	cupboard3=WTmovnode_load(SepCup3,"cupboard3.3ds",1.0f);
	pos[0]=-46.0f; pos[1]=0.0f; pos[2]=92.0f+891.0f;
	WTnode_settranslation(cupboard3,pos);
	WTnode_axisrotation(cupboard3,Y,(float)1.571,WTFRAME_LOCAL);

	instance=WTmovnode_instance(SepCup3,cupboard3);
	pos[0]=-46.0f; pos[1]=0.0f; pos[2]=92.0f+983.0f;
	WTnode_settranslation(instance,pos);
	WTnode_axisrotation(instance,Y,(float)1.571,WTFRAME_LOCAL);

	SepChair=WTsepnode_new(facility);
	chair=WTmovnode_load(SepChair,"chair.3ds",1.0f);
	pos[0]=-56.0-300.0f; pos[1]=0.0f; pos[2]=60.0f+420.0f;
	WTnode_settranslation(chair,pos);
	WTnode_axisrotation(chair,Y,(float)1.571,WTFRAME_LOCAL);
	Create_Instances(SepChair,chair,pos,5,4,56,60,1.571);

	SepCom=WTsepnode_new(facility);
	computer=WTmovnode_load(SepCom,"computer.3ds",1.0f);
	pos[0]=-47.0f; pos[1]=0.0f; pos[2]=64.0f+497.0f;
	WTnode_settranslation(computer,pos);
	WTnode_axisrotation(computer,Y,(float)1.571,WTFRAME_LOCAL);
	Create_Instances(SepCom,computer,pos,1,2,0,170,1.571);

	SepWash=WTsepnode_new(facility);
	washplace=WTmovnode_load(SepWash,"washplace.3ds",1.0f);
	pos[0]=-92.0f-404.0f; pos[1]=0.0f; pos[2]=0.0f+1166.0f;
	WTnode_settranslation(washplace,pos);
	WTnode_axisrotation(washplace,Y,(float)0.0,WTFRAME_LOCAL);

	SepRobTab=WTsepnode_new(facility);
	RobTab=WTmovnode_load(SepRobTab, "RobTab.3ds", 1.0f);
	pos[0]=-1000.0f; pos[1]=0.0f; pos[2]=120.0+112.0f;
	WTnode_settranslation(RobTab,pos);
	WTnode_axisrotation(RobTab,Y,(float)3.142,WTFRAME_LOCAL);

	SepRob1=WTsepnode_new(facility);
	Robot1=WTmovnode_load(SepRob1, "robot1.3ds", 1.0f);
	pos[0]=-10.0-1000.0f; pos[1]=-75.0f; pos[2]=212.0f;
	WTnode_settranslation(Robot1,pos);
	WTnode_axisrotation(Robot1,Y,(float)-1.5708, WTFRAME_LOCAL);
	SepRob2=WTsepnode_new(facility);
```

```
        Robot2=WTmovnode_load(SepRob2, "robot2.3ds", 1.0f);
        pos[0]=-20.0-1080.0f; pos[1]=-75.0f; pos[2]=112.0f;
        WTnode_settranslation(Robot2,pos);
        WTnode_axisrotation(Robot2,Y,(float)0.0,WTFRAME_LOCAL);

        SepRob3=WTsepnode_new(facility);
        Robot3=WTmovnode_load(SepRob3, "robot3.3ds", 1.0f);
        pos[0]=-20.0-1240.0f; pos[1]=0.0f; pos[2]=20.0+472.0f;
        WTnode_settranslation(Robot3,pos);
        WTnode_axisrotation(Robot3,Y,(float)1.5708,WTFRAME_LOCAL);

        SepCnc1=WTsepnode_new(facility);
        Cnc1=WTmovnode_load(SepCnc1, "cnc1.3ds", 1.0f);
        pos[0]=-1200.0f; pos[1]=0.0f; pos[2]=300.0f;
        WTnode_settranslation(Cnc1,pos);
        WTnode_axisrotation(Cnc1,Y,(float)0.5236,WTFRAME_LOCAL);

        SepCnc2=WTsepnode_new(facility);
        Cnc2=WTmovnode_load(SepCnc2, "cnc2.3ds", 1.0f);
        pos[0]=-1200.0f; pos[1]=0.0f; pos[2]=680.0f;
        WTnode_settranslation(Cnc2,pos);
        WTnode_axisrotation(Cnc2,Y,(float)3.142-0.5236f,WTFRAME_LOCAL);

        SepConveyor=WTsepnode_new(facility);
        Conveyor=WTmovnode_load(SepConveyor,"Conveyor.3ds", 1.0f);
        pos[0]=-300.0-820.0f; pos[1]=0.0f; pos[2]=510.0+262.0f;
        WTnode_settranslation(Conveyor,pos);
        WTnode_axisrotation(Conveyor,Y,(float)1.5708,WTFRAME_LOCAL);

        SepCim=WTsepnode_new(facility);
        Cim=WTmovnode_load(SepCim, "cim.3ds", 1.0f);
        pos[0]=1.0f; pos[1]=0.0f; pos[2]=0.0f;
        WTnode_settranslation(Cim,pos);
};

void Create_Instances(WTnode *root, WTnode* base, WTp3 &pos, int numX,
   int numY, int DistX, int DistY, double rotation)
{
        WTp3 p;
        WTp3_init(p);

        for(int x=0; x<numX; x++)
        {
                for(int y=0; y<numY; y++)
                {
                        if((x==0) && (y==0))
                        {
                                p[0]=pos[0]; p[1]=pos[1]; p[2]=pos[2];
                                continue;
                        }
                        instance=WTmovnode_instance(root,base);
                        p[2]=p[2]+DistY;
                        WTnode_settranslation(instance, p);

                        WTnode_axisrotation(instance,Y,(float)rotation,
                           WTFRAME_LOCAL);
                }


                p[2]=pos[2]-DistY;
                p[0]=p[0]-DistX;
        }
};

void Actions(void)
{
        short key;
```

```
key=WTkeyboard_getkey();

switch(con)
{
case 0:
        break;
case 1:

        {
                unsigned int n=0;
                unsigned char R=0, G=0, B=0, A=0;

                //Windows image capture
                unsigned char image[MapWidth*MapHeight*4];
                int temp[MapWidth*MapHeight];

                //Capturing for the Top View

                WTwindow_getimage(Top,0,210,MapWidth,MapHeight,
                   image);

                n=0;
                for(i=0; i<MapHeight; i++)
                {
                        for(unsigned int j=0; j<MapWidth; j++)
                        {
                                R=image[n];
                                n++;
                                G=image[n];
                                n++;
                                B=image[n];
                                n++;
                                A=image[n];
                                n++;
                                if((R==161) && (G==161) && (B==161) &&
                                   (A==255))
                                {
                                        temp[i*MapWidth+j]=1;
                                }
                                else
                                {
                                        temp[i*MapWidth+j]=9;
                                };
                        }
                }

                //Manipulation of Windows for VE
                WTwindow_delete(Top);

                Path_Planning(temp);
                SepMobileR=WTsepnode_new(facility);

                MobileR=WTmovnode_load(SepMobileR, "MobileR.3ds",
                   1.0f);
                WTnode_axisrotation(MobileR,Y,(float)1.5708,
                  WTFRAME_LOCAL);
                Loop=0; con=2;
                break;
        }
case 2:
        {
                Loop++;
                if(Loop==1)
                {
                        timespan=0;
                }
                timespan=timespan+0.1;
```

```
                    if(Final->GetTraj(MR,rotation,timespan))
                    {
                        WTnode_gettranslation(MobileR,test);
                        WTnode_getorientation(MobileR,viewrot);
                        WTnode_settranslation(MobileR,MR);
                        WTnode_axisrotation(MobileR,Y,(float)rotation,
                          WTFRAME_LOCAL);
                        pos[0]=MR[0]; pos[1]=-100.0f; pos[2]=MR[2];
                        WTviewpoint_setposition(camera,pos);
                        WTviewpoint_setorientation(camera,viewrot);
                    }
                    break;
            }
};

switch(tolower(key))
{
case 'q':
        WTuniverse_stop();
        break;
case WTKEY_F1:
        WTwindow_setprojection(MainWin,
          WTPROJECTION_SYMMETRIC);
        pos[0]=-750.0f; pos[1]=-100.0f; pos[2]=300.0f;
        WTviewpoint_setposition(camera1,pos);
        WTwindow_setviewpoint(MainWin,camera1);
        break;
case WTKEY_F2:
        posTop[0]=-748.0f; posTop[1]=-904.0f; posTop[2]=617.0f;
        WTviewpoint_setposition(TopView,posTop);
        WTwindow_setprojection(MainWin,
          WTPROJECTION_ORTHOGRAPHIC);
        WTwindow_zoomviewpoint(MainWin);
        WTwindow_setviewpoint(MainWin,TopView);
        break;
case WTKEY_F3:
        camera=WTviewpoint_new();
        WTwindow_setprojection(MainWin,
          WTPROJECTION_SYMMETRIC);
        WTviewpoint_setposition(camera,pos);
        WTviewpoint_setorientation(camera,viewrot);
        WTwindow_setviewpoint(MainWin,camera);
        break;

case WTKEY_END:
        Top=WTwindow_new(0,0,415,374,WTWINDOW_DEFAULT|
          WTWINDOW_NOBORDER);
        WTwindow_setviewpoint(Top,TopView);
        WTwindow_setprojection(Top,
          WTPROJECTION_ORTHOGRAPHIC);
        WTwindow_zoomviewpoint(Top);
        posTop[0]=62.0f; posTop[1]=-1000.0f; posTop[2]=-178.0f;
        WTviewpoint_setposition(TopView,posTop);
        con=1;
        break;
case WTKEY_LEFTARROW:
        posTop[0]=posTop[0]+1;
        WTviewpoint_setposition(TopView,posTop);
        break;
case WTKEY_RIGHTARROW:
        posTop[0]=posTop[0]-1;
        WTviewpoint_setposition(TopView,posTop);
        break;
case WTKEY_DOWNARROW:
        posTop[2]=posTop[2]+1;
        WTviewpoint_setposition(TopView,posTop);
```

```
                break;
        case WTKEY_UPARROW:
                posTop[2]=posTop[2]-1;
                WTviewpoint_setposition(TopView,posTop);
                break;
        }
};

void Path_Planning(int *CMap)
{
        int xi=0, xg=0, yi=0, yg=0, meetxi=0, meetyi=0, meetxg=0, meetyg=0;
        unsigned int counter=0;
        unsigned int steps=0;
        Astar *astarsearch=new Astar;

        //Obtaining Path Searching Information
        cout<<"The coordinate for initial point\n"<<endl;
        cout<<"Initial x: ";
        cin>>meetxi;
        cout<<"Initial y: ";
        cin>>meetyi;

        //Create Goal Node
        cout<<"\nThe coordinate for goal point\n"<<endl;
        cout<<"Goal x: ";
        cin>>meetxg;
        cout<<"Goal y: ";
        cin>>meetyg;

        xi=meetxi/7.5;
        xg=meetxg/7.5;
        yi=meetyi/7.5;
        yg=meetyg/7.5;

        //C-Space Generation
        astarsearch->Mapping(xi,yi,xg,yg,CMap);


        //Set Start and Goal States in Astar Search Algorithm
        State SearchState=SEARCH_STATE_SEARCHING;
        unsigned int SearchLoop=0;
        astarsearch->SetStartAndGoalState(xi,xg,yi,yg);

        //Start with the Graph Searching
        do
        {
                SearchState=astarsearch->SearchRule();
                SearchLoop++;
        } while(SearchState == SEARCH_STATE_SEARCHING);

        //Start of the solution path extraction
        if(SearchState == SEARCH_STATE_SUCCEEDED)
        {
                Astar::SearchNodeProperty *SolvedPath = astarsearch
                   ->GetSolutionStart();
                if(!SolvedPath)
                {
                        exit(0);
                };

                //Adding Solved Path into Another Object
                Final->AddSolvedNode(SolvedPath->x, SolvedPath->y, steps);
                SolvedPath->NodeInfo();
                for(;;)
                {
                        SolvedPath=astarsearch->GetSolutionNext();
```

```
                        if(!SolvedPath)
                        {
                                break;
                        }
                        steps++;
                        Final->AddSolvedNode(SolvedPath->x, SolvedPath->y, steps);

                        SolvedPath->NodeInfo();
                };
                cout<<"Solution steps: "<<steps<<endl;
        }
        else if(SearchState == SEARCH_STATE_FAILED)
        {
                cout<<"Search terminated -> Goal state failed to be found\n"<<endl;
                exit(0);
        };
        cout<<"Vertices Visited: "<<SearchLoop<<endl;

        //Path Refining to Remove 'cusps'
        Final->Repath(astarsearch);

        //The Generation of Time-based Trajectory from Geometrical Path
        Final->Trajectory();

        //Delete the Object for Path Planning to Free Up the Memory
        delete astarsearch;
};
```

Appendix C

Program Modules for the Experimental Mobile Manipulator

The most important three program modules in the program, i.e., *main()*, *ReadMobileManipulatorSensors()*, and *MobileManipulator_RAC_KBFAFC_RUN()* are shown in **Figures C.1, C.2** and **C.3** respectively.

```
int main(void)
{
  clrscr;
  printf("\n\n                          \n");
  printf("  Welcome to RTM MobileManipulator Online System\n");
  printf("     Version 1.852 (c)2005 by Endra Pitowarno\n\n");
  printf("            Please wait for a moment\n");
  printf("         The system is being calibrated\n\n");

  SystemCalibration();   Display_Init();
  Motor_Init();   MMotor_Init();

  MobileManipulator_RAC_AFC_RUN();
  MobileManipulator_RAC_KBFAFC_RUN();
  MobileManipulatorREMOTE();

  CloseOperation();
  return 0;
}
```

Figure C.1: Program module *main()*.

As can be seen in **Figure C.1**, the first routine executed in the program is *SystemCalibration()* followed by the initializations. The RACAFC and RACKBFAFC schemes are executed consecutively. A routine of *MobileManipulatorREMOTE()* was added to the end of program to allow user to control (move) the robot manually by remote system using the keyboard of PC. This is important for adjusting the initial (angle) position of the arm when the system starts to be executed.

```
float ReadMobileManipulatorSensors()
{
    Link1_DIR = COM_DIR & 0x01;
    Link2_DIR = COM_DIR & 0x02;
    LinkL_DIR = MCOM_DIR & KeepML;
    LinkR_DIR = MCOM_DIR & KeepMR;

    if (LinkL_DIR == ML_reverse)
            {
              th_d_L_act = -(MRead_ADC2()-cal_MADC2)*ML_th_d_Calib;
              gettime(&t); tsequel_L_rev = t.ti_min*60. + t.ti_sec+t.ti_hund/100.-subtt;
              th_L_act = th_L_act - fabs(th_d_L_act*(tsequel_L_rev-tsequel_L_rev_OLD));
              th_dd_L_act = -(MRead_ADC4()*1.-cal_MADC4)/20480.;
              Ic_L = -(fabs(MRead_ADC6()*1.-cal_MADC6))/2048.*50.;
            }
    if (LinkL_DIR == ML_forward)
            {
              th_d_L_act = (MRead_ADC2()-cal_MADC2)*ML_th_d_Calib;
              gettime(&t); tsequel_L_for = t.ti_min*60. + t.ti_sec+t.ti_hund/100.-subtt;
              th_L_act = th_L_act + th_d_L_act*(tsequel_L_for-tsequel_L_for_OLD);
              th_dd_L_act = (MRead_ADC4()*1.-cal_MADC4)/20480.;
              Ic_L = fabs(MRead_ADC6()*1.-cal_MADC6)/2048.*50.;
            }
    if (LinkR_DIR == MR_forward)
            {
              th_d_R_act = (MRead_ADC1()-cal_MADC1)*MR_th_d_Calib;
              gettime(&t); tsequel_R_for = t.ti_min*60. + t.ti_sec+t.ti_hund/100.-subtt;
              th_R_act = th_R_act + th_d_R_act*(tsequel_R_for-tsequel_R_for_OLD);
              th_dd_R_act = (MRead_ADC3()*1.-cal_MADC3)/20480.;
              Ic_R = fabs(MRead_ADC5()*1.-cal_MADC5)/2048.*50.;
            }
    if (LinkR_DIR == MR_reverse)
            {
              th_d_R_act = -(MRead_ADC1()-cal_MADC1)*MR_th_d_Calib;
              gettime(&t); tsequel_R_rev = t.ti_min*60. + t.ti_sec+t.ti_hund/100.-subtt;
              th_R_act = th_R_act - fabs(th_d_R_act*(tsequel_R_rev-tsequel_R_rev_OLD));
              th_dd_R_act = -(MRead_ADC3()*1.-cal_MADC3)/20480.;
              Ic_R = -(fabs(MRead_ADC5()*1.-cal_MADC5))/2048.*50.;
            }

    tsequel_L_rev_OLD = tsequel_L_rev;
    tsequel_L_for_OLD = tsequel_L_for;
    tsequel_R_rev_OLD = tsequel_R_rev;
    tsequel_R_for_OLD = tsequel_R_for;
```

<page 1 of 2 of Figure C.2> continued...

```
    PHI_d_act = -rWheel/(2.*halfB)*th_d_L_act+rWheel/(2.*halfB)*th_d_R_act;
    PHI_act = PHI_act + PHI_d_act*(tt-tsequel_PHI_OLD);
    tsequel_PHI_OLD = tt;

    XB_d_act = th_d_L_act*(rWheel/2.*cos(PHI_act)+dGtoF*rWheel/(2.*halfB)*sin(PHI_act))
            + th_d_R_act*(rWheel/2.*cos(PHI_act)-dGtoF*rWheel/(2.*halfB)*sin(PHI_act));
    YB_d_act = th_d_L_act*(rWheel/2.*sin(PHI_act)-dGtoF*rWheel/(2.*halfB)*cos(PHI_act))
            + th_d_R_act*(rWheel/2.*sin(PHI_act)+ dGtoF*rWheel/(2.*halfB)*cos(PHI_act));
    XB_act = XB_act + XB_d_act*(tt-tsequel_XYB_act_OLD);
    YB_act = YB_act + YB_d_act*(tt-tsequel_XYB_act_OLD);
    tsequel_XYB_act_OLD = tt;

    /****************************************** Manipulator's Sensors ***/
    th_1_act = ReadEncoderSATU()/4096.*2*PI;
    th_2_act = ReadEncoderDUA()/4096.*2*PI;

    if (Link1_DIR == M1_CW)
            {
        th_d_1_act = -(fabs(Read_ADC3()*1.-cal_ADC3*1.)*th_d_1_calib);
              th_dd_1_act = fabs(Read_ADC5()*1.-cal_ADC5*1.)/2048.*cal_thdd1;
              Ic_1 = Sample_Ic_1/200.;
            }

    if (Link1_DIR == M1_CCW)
            { th_d_1_act = fabs(Read_ADC3()*1.-cal_ADC3*1.)*th_d_1_calib;
              th_dd_1_act = -(fabs(Read_ADC5()*1.-cal_ADC5*1.))/2048.*cal_thdd1;
              Ic_1 = Sample_Ic_1/200.;
            }

    if (Link2_DIR == M2_CW)
            {
              th_d_2_act = -(fabs(Read_ADC4()*1.-cal_ADC4*1.)*th_d_2_calib);
              th_dd_2_act = fabs(Read_ADC6()*1.-cal_ADC6*1.)/2048.*cal_thdd2;
              Ic_2 = Sample_Ic_2/400.;
            }

    if (Link2_DIR == M2_CCW)
            {
              th_d_2_act = fabs(Read_ADC4()*1.-cal_ADC4*1.)*th_d_2_calib;
              th_dd_2_act = -(fabs(Read_ADC6()*1.-cal_ADC6*1.))/2048.*cal_thdd2;
              Ic_2 = Sample_Ic_2/400.;
            }

    /*************************************************************/
    return(th_R_act,th_L_act,
                th_1_act,th_2_act,
                th_d_R_act,th_d_L_act,
                th_d_1_act,th_d_2_act,
                th_dd_R_act,th_dd_L_act,
                th_dd_1_act,th_dd_2_act,
                Ic_1,Ic_2,Ic_L,Ic_R,
                PHI_act,PHI_d_act,
                XB_act,YB_act,XB_d_act,YB_d_act);
}
```

<page 2 of 2 of Figure C.2>

Figure C.2: Program module *ReadMobileManipulatorSensors()*.

In **Figure C.2**, it can be seen that the sign of sensors value of velocity and acceleration is obtained by combining the direction of motor's moving according to the sensors value. If the direction is clock-wise (CW), then the sensor value is converted to negative vice versa.

```
/************** Integrated MOBILE MANIPULATOR SECTION (RAC-KBFAFC) ****/

int MobileManipulator_RAC_KBFAFC_RUN()
{
  i = 0; ii = 0; th_R_des = 0; th_L_des = 0;
  PHI_des_tt = 0; tsequel_th_RL_des_OLD = 0; PHI_act = 0;
  XB_act = 0; YB_act = 0; TEc = 0; tt = 0; th_L_act = 0;   th_R_act = 0;

  data_t_Rec = fopen("d:/endra/kt_MM.dat","w+");
  data_TE_Arm_Rec = fopen("d:/endra/kTE_Arm.dat","w+");
  data_TE_Body_Rec = fopen("d:/endra/kTE_Body.dat","w+");
  data_X_des_Rec = fopen("d:/endra/kX_des.dat","w+");
  data_Y_des_Rec = fopen("d:/endra/kY_des.dat","w+");
  data_XB_des_Rec = fopen("d:/endra/kXB_des.dat","w+");
  data_YB_des_Rec = fopen("d:/endra/kYB_des.dat","w+");
  data_X_act_Rec = fopen("d:/endra/kX_act.dat","w+");
  data_Y_act_Rec = fopen("d:/endra/kY_act.dat","w+");
  data_XB_act_Rec = fopen("d:/endra/kXB_act.dat","w+");
  data_YB_act_Rec = fopen("d:/endra/kYB_act.dat","w+");
  data_th_dd_1_act_Rec = fopen("d:/endra/kth_dd_1_act.dat","w+");
  data_th_dd_2_act_Rec = fopen("d:/endra/kth_dd_2_act.dat","w+");
  data_Ic_1_Rec = fopen("d:/endra/kIc_1.dat","w+");
  data_Ic_2_Rec = fopen("d:/endra/kIc_2.dat","w+");

  Display_Sys_MM();
  Display_ProgramCode();
  RAFCON_LOGO_RACKBFAFC();
  SetControlParametersMM();
  Set_Intelligent_KBF_Parameter();
  MobileManipulatorREMOTE();
  ManipulatorHomePosition();
  MobileManipulatorREMOTE();
  MessagesBOTTOM_Ready();
  MessagesBOTTOM_RUN();

  PHI_d_des = -rWheel/(2.*halfB)*th_d_L_des+rWheel/(2.*halfB)*th_d_R_des;
  PHI_act = InitialOrientationPHI;
  SetTimeRUN();   GetCurrentTime();

  for(;;)
  {
   gettime(&t);       tt = t.ti_min*60. + t.ti_sec+t.ti_hund/100.-subtt;
   ReadMobileManipulatorSensors();
   RealTimeDisplay();
   Tq1 = Ic_1*Ktn1; Tq2 = Ic_2*Ktn2;
   KVkbf1 = KBF_module_1(); KVkbf2 = KBF_module_2();
   ActionKBFAFC1 = (Tq1-th_dd_1_act*KVkbf1) / Ktn1;
   ActionKBFAFC2 = (Tq2-th_dd_2_act* KVkbf2) / Ktn2;
```

<page 1 of 4 of Figure C.3> continued...

```
/****************************** Manipulator RUN Section ************************/
if (th_1_act > th_1_des)
 {
        Status_DIR = COM_DIR & KeepM2;  COM_DIR = (Status_DIR + M1_CW);
        outp(DIG_OUT_Address,COM_DIR);
        ActionR = (fabs( Kp1*(th_1_des-th_1_act) +
                  Kd1*(th_d_1_des-th_d_1_act))*(AFCCA_Value1+KVkbf1)/Ktn1 +
                  ActionKBFAFC1)*Ktn1*Kgearbox1+M1_Calib;
        if(ActionR>=4000) ActionR=4000;
        ActionH = (ActionR/16);       ActionL = (ActionR-floor(ActionH*16));
        outp(DAC_Motor1_H,ActionH);          outp(DAC_Motor1_L,ActionL);
 }
if (th_1_act < th_1_des )
 {
        Status_DIR = COM_DIR & KeepM2;  COM_DIR = (Status_DIR + M1_CCW);
        outp(DIG_OUT_Address,COM_DIR);
        ActionR = (fabs( Kp1*(th_1_des-th_1_act) +
                  Kd1*(th_d_1_des-th_d_1_act))*(AFCCA_Value1+KVkbf1)/Ktn1 +
                  ActionKBFAFC1)*Ktn1*Kgearbox1+M1_Calib;
        if(ActionR>=4000) ActionR=4000;
        ActionH = ActionR/16;   ActionL = ActionR-floor(ActionH*16);
        outp(DAC_Motor1_H,ActionH);          outp(DAC_Motor1_L,ActionL);
        outp(PPIPA,ActionH);
 }
if (th_2_act > th_2_des)
 {
        Status_DIR = COM_DIR & KeepM1;   COM_DIR = (Status_DIR + M2_CW);
        outp(DIG_OUT_Address,COM_DIR);
        ActionR = (fabs( Kp2*(th_2_des-th_2_act) +
                  Kd2*(th_d_2_des-th_d_2_act))*(AFCCA_Value2+KVkbf2)/Ktn2 +
                  ActionKBFAFC2)*Ktn2*Kgearbox2+M2_Calib;
        if(ActionR>=4000) ActionR=4000;
        ActionH = ActionR/16;   ActionL = ActionR-floor(ActionH*16);
        outp(DAC_Motor2_H,ActionH);   outp(DAC_Motor2_L,ActionL);
 }
if (th_2_act < th_2_des )
 {
        Status_DIR = COM_DIR & KeepM1;   COM_DIR = (Status_DIR + M2_CCW);
        outp(DIG_OUT_Address,COM_DIR);
        ActionR = (fabs( Kp2*(th_2_des-th_2_act) +
                  Kd2*(th_d_2_des-th_d_2_act))*(AFCCA_Value2+KVkbf2)/Ktn2 +
                  ActionKBFAFC2)*Ktn2*Kgearbox2+M2_Calib;
        if(ActionR>=4000) ActionR=4000;
        ActionH = ActionR/16;   ActionL = ActionR-floor(ActionH*16);
        outp(DAC_Motor2_H,ActionH);   outp(DAC_Motor2_L,ActionL);
 }

/************************ Mobile Platform RUN Section **************************/
 if (th_L_act > th_L_des)
  {
    MStatus_DIR = MCOM_DIR & KeepMR; MCOM_DIR = (MStatus_DIR + ML_reverse);
        outp(MDIG_OUT_Address,MCOM_DIR);
        MActionR = fabs( KpL*(th_L_des-th_L_act) + KdL*(th_d_L_des-th_d_L_act))
                        *KtnL*KgearboxL+ML_Calib;
        if(MActionR>=4000) MActionR=4000;
        MActionH = MActionR/16;   MActionL = (MActionR-floor(MActionH*16));
        outp(MDAC2_MotorL_H,MActionH);   outp(MDAC2_MotorL_L,MActionL);
 }
```

<page 2 of 4 of Figure C3> continued...

```
if (th_L_act < th_L_des )
 {
    MStatus_DIR = MCOM_DIR & KeepMR; MCOM_DIR = (MStatus_DIR + ML_forward);
      outp(MDIG_OUT_Address,MCOM_DIR);
      MActionR = fabs( KpL*(th_L_des-th_L_act) + KdL*(th_d_L_des-th_d_L_act))
                    *KtnL*KgearboxL+ML_Calib;
      if(MActionR>=4000) MActionR=4000;
      MActionH = MActionR/16;  MActionL = MActionR-floor(MActionH*16);
      outp(MDAC2_MotorL_H,MActionH);  outp(MDAC2_MotorL_L,MActionL);
      outp(MPPIPA,MActionH);
 }
if (th_R_act > th_R_des)
 {
    MStatus_DIR = MCOM_DIR & KeepML; MCOM_DIR = (MStatus_DIR + MR_reverse);
      outp(MDIG_OUT_Address,MCOM_DIR);
      MActionR = fabs( KpR*(th_R_des-th_R_act) + KdR*(th_d_R_des-th_d_R_act))
                    *KtnR*KgearboxR+MR_Calib;
      if(MActionR>=4000) MActionR=4000;
      MActionH = MActionR/16;  MActionL = MActionR-floor(MActionH*16);
      outp(MDAC1_MotorR_H,MActionH); outp(MDAC1_MotorR_L,MActionL);
 }
if (th_R_act < th_R_des )
 {
    MStatus_DIR = MCOM_DIR & KeepML; MCOM_DIR = (MStatus_DIR + MR_forward);
      outp(MDIG_OUT_Address,MCOM_DIR);
      MActionR = fabs( KpR*(th_R_des-th_R_act) + KdR*(th_d_R_des-th_d_R_act))
                    *KtnR*KgearboxR+MR_Calib;
      if(MActionR>=4000) MActionR=4000;
      MActionH = MActionR/16;  MActionL = MActionR-floor(MActionH*16);
      outp(MDAC1_MotorR_H,MActionH); outp(MDAC1_MotorR_L,MActionL);
 }
if (tt >= Tstop)
 {
      outp(DAC_Motor1_H,0x00); outp(DAC_Motor1_L,0x00);
      outp(DAC_Motor2_H,0x00); outp(DAC_Motor2_L,0x00);
      outp(MDAC1_MotorR_H,0x00); outp(MDAC1_MotorR_L,0x00);
      outp(MDAC2_MotorL_H,0x00); outp(MDAC2_MotorL_L,0x00);

      int j = ii;
      for (i=0; i<j; i++)
      {
       fprintf(data_t_Rec, "%10f \n",t_Rec[i]);
       fprintf(data_TE_Arm_Rec, "%10f \n",TE_Arm_Rec[i]);
       fprintf(data_TE_Body_Rec, "%10f \n",TE_Body_Rec[i]);
       fprintf(data_X_des_Rec, "%10f \n",X_des_Rec[i]);
       fprintf(data_Y_des_Rec, "%10f \n",Y_des_Rec[i]);
       fprintf(data_XB_des_Rec, "%10f \n",XB_des_Rec[i]);
       fprintf(data_YB_des_Rec, "%10f \n",YB_des_Rec[i]);
       fprintf(data_X_act_Rec, "%10f \n",X_act_Rec[i]);
       fprintf(data_Y_act_Rec, "%10f \n",Y_act_Rec[i]);
       fprintf(data_XB_act_Rec, "%10f \n",XB_act_Rec[i]);
       fprintf(data_YB_act_Rec, "%10f \n",YB_act_Rec[i]);
       fprintf(data_th_dd_1_act_Rec, "%10f \n",th_dd_1_act_Rec[i]);
       fprintf(data_th_dd_2_act_Rec, "%10f \n",th_dd_2_act_Rec[i]);
       fprintf(data_Ic_1_Rec, "%10f \n",Ic_1_Rec[i]);
       fprintf(data_Ic_2_Rec, "%10f \n",Ic_2_Rec[i]);
      }
```

<page 3 of 4 of Figure C3> continued...

```
            fclose(data_YB_des_Rec);   fclose(data_X_act_Rec);   fclose(data_Y_act_Rec);
            fclose(data_t_Rec);   fclose(data_TE_Arm_Rec);   fclose(data_TE_Body_Rec);
            fclose(data_X_des_Rec);   fclose(data_Y_des_Rec);   fclose(data_XB_des_Rec);
            fclose(data_XB_act_Rec);   fclose(data_YB_act_Rec);   fclose(data_th_dd_1_act_Rec);
            fclose(data_th_dd_2_act_Rec);   fclose(data_Ic_1_Rec);   fclose(data_Ic_2_Rec);

          setviewport(x/2+8,308,x/2+285,320,CLIP_ON); clearviewport();
          sprintf(msg," RACKBFAFC : Check the Results !",10); outtextxy(0,2,msg);
          getch();  return 0;
     }

     jj=jj+1;
     if(jj>=4)
     {
          tt_old = tt;
          t_Rec[ii] = tt; TE_Arm_Rec[ii] = TErA; TE_Body_Rec[ii] = TEr;
          X_des_Rec[ii] = XdesV; Y_des_Rec[ii] = YdesV; XB_des_Rec[ii] = XB_des;
          YB_des_Rec[ii] = YB_des; X_act_Rec[ii] = XactV; Y_act_Rec[ii] = YactV;
          XB_act_Rec[ii] = XB_act; YB_act_Rec[ii] = YB_act;
          th_dd_1_act_Rec[ii] = th_dd_1_act;
          th_dd_2_act_Rec[ii] = th_dd_2_act; Ic_1_Rec[ii] = Ic_1; Ic_2_Rec[ii] = Ic_2;
          ii++; jj=0;
     }

    /*************************** emergency STOP ****************************/
    while(kbhit())
                    {
                     VR = MRead_ADC7();
                     if(VR < 2990) sound(1200);
                     VL = MRead_ADC8();
                     if(VL < 2990) sound(800);
                     outp(DAC_Motor1_H,0x00);  outp(DAC_Motor1_L,0x00);
                     outp(DAC_Motor2_H,0x00);  outp(DAC_Motor2_L,0x00);
                     outp(MDAC1_MotorR_H,0x00);  outp(MDAC1_MotorR_L,0x00);
                     outp(MDAC2_MotorL_H,0x00);  outp(MDAC2_MotorL_L,0x00);

                     getch();  nosound();
                     return 0;
                    }
  }
}
```

<page 4 of 4 of Figure C.3>

Figure C.3: Program module MobileManipulator_RAC_KBFAFC_RUN().

The program module *MobileManipulator_RAC_KBFAFC_RUN()* as shown in **Figure C.3** contains three main parts that are manipulator control loop, mobile platform control loop, and Data Recording/Saving routines. Note that in this program, the RACKBFAFC scheme was applied to the manipulator section only, while RACAFC was applied to the platform. An emergency *STOP* facility was implemented at the end of the program as a safety precautionary measure. Users can immediately stop the program execution if a fault operation occurs by hitting any key of the keyboard when the robot is running.

Appendix D

List of Publications

1. Tang, H. H., Mailah, M. and Kasim, M. 'Stabilization of Nonholonomic Wheeled Mobile Robot through Intelligent Active Force Control'. *Proceedings of Advanced Technology Congress – Intelligent Systems and Robotics (CISAR 2003)*. May 20-21, 2003. Kuala Lumpur, Malaysia: Institute of Advanced Technology, UPM. 2003.
2. Tang, H. H., Mailah, M. and Kasim, M. Collision-Free Global Path Planning for a Holonomic Mobile Robot in a Known Stationary Virtual Environment. *Proceedings of Malaysian Science and Technology (MSTC) – Information and Communication Technology*. September 23-25, 2003. Kuala Lumpur, Malaysia: MSTC. 2003. 328-335.
3. Didik Setyo Purnomo and Musa Mailah, 'Trajectory Tracking of Nonholonomic Mobile Robot Using Intelligent Active Force Control', *Procs. of Advanced Technology Congress 2003*, Marriot Hotel, Kuala Lumpur, May 2003.
4. Didik Setyo Purnomo and Musa Mailah, 'Control of Nonholonomic Mobile Robot Using Adaptive Active Force Control Strategy', *Procs. of CARS-FOF 2003*, SIRIM, Kuala Lumpur, July 2003.
5. Didik S.P., Musa Mailah, 'Optimization of The Membership Function in A Fuzzy Logic Based Active Force Control Applied To A Wheeled Mobile Robot', *Procs. of MSTC2003*, Hotel Cititel, Kuala Lumpur, 23-26 September 2003.
6. Endra Pitowarno, Musa Mailah, 'A Disturbance Cancellation Method Of A Differentially Driven Non-Holonomic Mobile Robot Using Active Force Control with Optimized Crude Approximation', *Procs. of MSTC2003*, Hotel Cititel, Kuala Lumpur, 23-26 September 2003.

7. Endra Pitowarno, **Musa Mailah**, 'A Robust Motion Control of A Differentially Driven Mobile Robot Using Resolved Acceleration and Active Force Control', *Procs. of IES 2003*, December 16, 2003, Surabaya, Indonesia.

8. Endra Pitowarno, Musa Mailah, Hishamuddin Jamaluddin 'Motion Control of Mobile Robot Using Resolved Acceleration and Active Force Control', Paper No. 128, ICCC 2004, 25-28 May 2004, Zakopane, Poland.

9. Didik Setyo Purnomo, Endra Pitowarno, Musa Mailah, 'Motion Control of A Nonholonomic Mobile Robot Using Fuzzy Logic Active Force Control', *Procs. of IES 2004*, October 2004, Surabaya, Indonesia.

10. Endra Pitowarno, Musa Mailah, Hishamuddin Jamaluddin, 'Motion Control of Mobile Manipulator Using Resolved Acceleration and Iterative Learning Active Force Control', *Procs. of ICOM05*, KL, May 2005.

11. Endra Pitowarno, Musa Mailah, 'Resolved Acceleration and Knowledge Based Fuzzy Active Force Control for Mobile Manipulator', *Procs. of ROVISP05*, USM Penang, July 2005.

12. Musa Mailah, Endra Pitowarno, Hishamuddin Jamaluddin, 'Robust Motion Control for Mobile Manipulator Using Resolved Acceleration and Proportional-Integral Active Force Control', *International Journal of Advanced Robotic Systems*, June 2005, Vol. 2, No.2, pp 125-134.

13. Tang Howe Hing, Musa Mailah and M.K. Abdul Jalil, 'Robust Intelligent Active Force Control of Nonholonomic Wheeled Mobile Robot', *Jurnal Teknologi (D)*, UTM, June 2006.

14. Musa Mailah, Tang Howe Hing, M Kasim A. Jalil, 'Virtual Wheeled Mobile Robot Simulator with Integrated Motion Planning and Active Force Control' in *Advanced Technologies, Research – Development – Application*, Ed. Bojan Lalic, Pro Literatur Verlag, Germany, ISBN 3-86611-197-5, September 2006, pp. 615-640.

15. Tang Howe Hing, Musa Mailah, 'Motion Control of Nonholonomic Wheeled Mobile Robot in A Structured Layout', *Jurnal Mekanikal*, June 2006.

16. Tang Howe Hing, Musa Mailah, M.K. Abdul Jalil 'Virtual Simulator for Control of Autonomous Nonholonomic Wheeled Mobile Robot', submitted to *Brazilian Journal of Mechanical Science & Engineering*, August 2006.

Appendix E

Achievements / Outputs / Beneficiaries / Awards

A. BOOK CHAPTER:

1. Musa Mailah, Tang Howe Hing, M Kasim A. Jalil, 'Virtual Wheeled Mobile Robot Simulator with Integrated Motion Planning and Active Force Control' in *Advanced Technologies, Research – Development – Application*, Ed. Bojan Lalic, Pro Literatur Verlag, Germany, ISBN 3-86611-197-5, September 2006, pp. 615-640.

B. INTERNATIONAL JOURNAL:

1. Musa Mailah, Endra Pitowarno, Hishamuddin Jamaluddin, 'Robust Motion Control for Mobile Manipulator Using Resolved Acceleration and Proportional-Integral Active Force Control', *International Journal of Advanced Robotic Systems*, June 2005, Vol. 2, No.2, pp 125-134.
2. Tang Howe Hing, Musa Mailah, M.K. Abdul Jalil 'Virtual Simulator for Control of Autonomous Nonholonomic Wheeled Mobile Robot', submitted to *Brazilian Journal of Mechanical Science & Engineering*, August 2006.

C. NATIONAL JOURNAL:

1. Tang Howe Hing, Musa Mailah and M.K. Abdul Jalil, 'Robust Intelligent Active Force Control of Nonholonomic Wheeled Mobile Robot', *Jurnal Teknologi (D)*, UTM, June 2006.
2. Tang Howe Hing, Musa Mailah, 'Motion Control of Nonholonomic Wheeled Mobile Robot in A Structured Layout', *Jurnal Mekanikal*, June 2006.

D. INTERNATIONAL CONFERENCE:

1. Tang, H. H., Mailah, M. and Kasim, M. 'Stabilization of Nonholonomic Wheeled Mobile Robot through Intelligent Active Force Control'. *Proceedings of Advanced Technology Congress – Intelligent Systems and Robotics (CISAR 2003)*. May 20-21, 2003. Kuala Lumpur, Malaysia: Institute of Advanced Technology, UPM. 2003.
2. Didik Setyo Purnomo and Musa Mailah, 'Trajectory Tracking of Nonholonomic Mobile Robot Using Intelligent Active Force Control', *Procs. of Advanced Technology Congress 2003*, Marriot Hotel, Kuala Lumpur, May 2003.

3. Didik Setyo Purnomo and Musa Mailah, 'Control of Nonholonomic Mobile Robot Using Adaptive Active Force Control Strategy', *Procs. of CARS-FOF 2003*, SIRIM, Kuala Lumpur, July 2003.
4. Endra Pitowarno, Musa Mailah, 'A Robust Motion Control of A Differentially Driven Mobile Robot Using Resolved Acceleration and Active Force Control', *Procs. of IES 2003*, December 16, 2003, Surabaya, Indonesia.
5. Endra Pitowarno, Musa Mailah, Hishamuddin Jamaluddin 'Motion Control of Mobile Robot Using Resolved Acceleration and Active Force Control', Paper No. 128, ICCC 2004, 25-28 May 2004, Zakopane, Poland.
6. Didik Setyo Purnomo, Endra Pitowarno, Musa Mailah, 'Motion Control of A Nonholonomic Mobile Robot Using Fuzzy Logic Active Force Control', *Procs. of IES 2004*, October 2004, Surabaya, Indonesia.
7. Endra Pitowarno, Musa Mailah, Hishamuddin Jamaluddin, 'Motion Control of Mobile Manipulator Using Resolved Acceleration and Iterative Learning Active Force Control', *Procs. of ICOM05*, KL, May 2005.
8. Endra Pitowarno, Musa Mailah, 'Resolved Acceleration and Knowledge Based Fuzzy Active Force Control for Mobile Manipulator', *Procs. of ROVISP05*, USM Penang, July 2005.

E. NATIONAL CONFERENCE:

1. Endra Pitowarno, Musa Mailah, 'A Disturbance Cancellation Method Of A Differentially Driven Non-Holonomic Mobile Robot Using Active Force Control with Optimized Crude Approximation', *Procs. of MSTC2003*, Hotel Cititel, Kuala Lumpur, 23-26 September 2003.
2. Tang, H. H., Mailah, M. and Kasim, M. Collision-Free Global Path Planning for a Holonomic Mobile Robot in a Known Stationary Virtual Environment. *Proceedings of Malaysian Science and Technology (MSTC) – Information and Communication Technology*. September 23-25, 2003. Kuala Lumpur, Malaysia: MSTC. 2003. 328-335.
3. Didik S.P., Musa Mailah, 'Optimization of The Membership Function in A Fuzzy Logic Based Active Force Control Applied To A Wheeled Mobile Robot', *Procs. of MSTC2003*, Hotel Cititel, Kuala Lumpur, 23-26 September 2003.

F. PHD STUDENT:

1. Endra Pitowarno, *Intelligent Active Force Control of A Mobile Manipulator*, November 2002-February 2006 (Completed).

G. MENG STUDENT:

1. Tang Howe Hing, *Implementation of Motion Planning and Active Force Control to A Virtual Wheeled Mobile Robot*, May 2002-July 2004 (Completed).
2. Didik Setyo Purnomo, *Design and Development of Intelligent Mobile Robot Using Active Force Control*, May 2002-December 2004 (Completed).

H. AWARD:

1. A Bronze Medal in Exposition Science, Technology and Innovation (ST&I 2004) for the project research, '*Autonomous Wheeled Mobile Manipulator Using Active Force Control*', 27-29 August 2004, Kuala Lumpur.

I. COLLABORATION:

- Collaboration with **Polis Di Raja Malaysia (PDRM)** on *Retrofit of Mobile Bomb Disposal Robot*, 2005-2006.
- A loaned and initially malfunctioned *Cyclops Mobile Robot* (a mobile bomb disposal unit) has been ***successfully*** retrofitted with the aid of five (5) PSM students and an Assistant Research Officer in 2006.

Appendix F

Mechanical Design and Production Drawings of Mobile Manipulator